Universität Leipzig

Institut für Informatik
Fakultät für Mathematik und Informatik
Abteilung Datenbanken

Development of an Audio-Classifier for Urban Sounds under consideration of the PATE framework

Master Thesis

Submitted by:
Paul Muschiol

Matriculation number:
3752386

Supervisor:
Prof. Dr. Erhard Rahm
Maja Schneider

Abstract

Audio data recorded in urban environments contain a lot of valuable information. Data analysts can use data collections to optimize city planning, delivery routes, traffic management, and many other applications. The recordings contain personal and private information, intentionally or not, on the people close to the recording devices. Personal information qualifies to identify a person independent from the content of the data. This can be people talking to each other on the recording or similar sound patterns appearing at the exact location and time every day. All data that is not made intentionally public must be considered private.
It is impossible to eradicate the personal information from the data or prove that there is no information meant to remain private. Legal requirements make it hard to gather such data at scale for analysis and machine learning. The *PATE* framework introduced an approach to learn from private data that is distributed among multiple data owners [68], [67]. It creates a machine learning model from private data while efficiently bounding privacy exposure. While most real-life datasets are multi-label, *PATE* only supports multi-class classification. The main goal of this work is to transfer the *PATE* framework to a more general definition that supports multi-label classification tasks. In the evaluation section, this new approach is verified on the *Audio Set* data, which contains more than two million recordings of audio events [29]. It is evaluated in this work can transfer the promising results of *PATE* to the *Audio Set* successfully. The created classifiers for urban sound events get evaluated for their prediction performance and privacy guarantee.

Contents

List of Figures

List of Tables

List of Abbreviations

AP	average precision
AUC	area under the curve
BB	Basic Block
BCE	binary cross-entropy
CDF	cumulative distribution function
CNN	convolutional neural network
DCASE	Detection and Classification of Acoustic Scenes and Events4
DP	Differential Privacy
(ϵ, δ)**-DP**	ϵ, δ-differential privacy
GAN	Generative Adversarial Networks
GDPR	General Data Protection Regulation
GNMax	gaussian noise max
GNThreshold	Gaussian noise threshold
GNThreshold Adaptive	Gaussian noise adaptive threshold
GS	Global sensitivity
kNN	k-nearest neighbors
mAP	mean average precision
P	Precisions
PATE	private aggregation of teacher ensembles
PCA	principal component analysis
PPML	Privacy-preserving machine learning
R	Recall
ReLu	rectified linear unit
RDP	Rényi differential privacy
ROC	Reciever Operating Characteristic
ScaDS	Scalable Data Service and Solutions
SGD	stochastic gradient descent
SVM	support vector machine

1 Introduction

1.1 Motivation

The importance of data for a company changes from necessity to an individual asset. A global system of vendors gathers, organizes, and exchanges data to create value in a data economy. Next to classical products and services, the collected or created data has a value within itself. Existing companies must use all their information to remain competitive against fast-growing competitors or rethink how data impacts their business model overall. New ventures with the only purpose to monetize data itself arise. With the capability to process high amounts of data, dropping prices to store data, unnoticed information becomes interesting.

Existing information or communication from suppliers that were ever error-prone is replaced by analytics. Instead of trusting delivery dates from suppliers, data is gathered along the whole supply chain. In retail and e-commerce, the data value chain extends to the customers and enables companies to gain new insights.

Valuable data can be of any type. Filled forms, internet cookies, metadata, technical performance data, audiovisual recordings, and many more qualify for analytical processing. Especially audiovisual data still holds many undiscovered potentials. It is close to how humans interact with the world using their senses. The sheer amount of available data makes it impossible to be processed by humans. This opens opportunities for ongoing improvements by research and industry in this area.

There are already powerful applications for processing this kind of data. Amazon has with its Alexa service a powerful voice-based question and answer tool. For this, the devices transmit speech commands of the customers to Amazon's data centers for processing. Tesla utilizes the data stream of various cameras around their cars to enable nearly self-driving capabilities. Different companies aim to simplify navigation in cities through authentic images of the places and streets that a person walks through. Up-to-date video conferencing tools modify pictures of the participants in real-time to blur a background. All those examples have in common that they utilize audio, visual, or audiovisual data to provide new services or assets. Those are just the most prominent use cases, while many more ideas arise daily.

Working with this kind of data has one common issue. Nobody knows what information is inside the data. While most other data is structured so that you can describe what information it might contain, unstructured data types like audiovisual or text can have nearly everything. This includes confidential, private, or personal data. The European law understands every information related to an identified or identifiable living individual as personal data. The General Data Protection Regulation (GDPR) allows the processing of such personal data on a high level only if the data subject has given consent to process the data, for contractual obligation with the data subject, for legal obligations, or in the public interest [30]. Those constraints make it very hard, if not impossible, to collect and process audiovisual on a large scale. As long as there are only economic or research interests, every individual with personal rights must be asked formally for consent with the processing.

The only exception here is if the data has been rendered anonymous. Personal data that is

de-identified, encrypted, or pseudonymized effectively does not fall into the scope of the GDPR. However, if there is still a way to re-identify the person behind the data, it is still considered personal information. Such anonymization must be irreversible to be considered adequate. For all organizations and institutions that aim to collect audiovisual data in public space, this is the only option within the European Union to comply with GDPR. Asking everyone that might appear within the data is rather impractical.
Within the "Data Economy 4 Advanced Logistics" project [64], the Scalable Data Service and Solutions (ScaDS) Competence Center of the University Leipzig and other partners built a platform for logistics service providers to share and use data. Cameras should collect audiovisual data next to sensor data that collects speed, location, temperature, and around delivery vehicles. For this, microphones and cameras are mounted onto delivery cars and bikes. With this information, machine learning models are trained, or other assets are created. Data analysts can enhance such analytical tools or assets traffic information, early identification of issues with the vehicle, or improved route maps. Sound maps quantifying sound pollution, for example, are currently created manually by cities to justify measures for infrastructure planning.
One big step to further use the collected audiovisual data for other purposes is to classify what can be seen or heard within a time frame. While it is common for a human to see and identify an object or listen and associate it with a known event, it is still a complex task for computers. This thesis creates an audio classifier for sound events in the city and environmental sounds.
Processing audiovisual data of delivery vehicles require dealing with privacy constraints. The only legal way to process data of individuals that do not consent with the processing or contractual permit is granted is to prove the anonymity of the data. Cleaning data from personal information is always a trade-off between data quality and privacy. While removing this data from the data set, the data itself may become unusable.
Nicolas Papernot et al. introduced a framework for private aggregation of teacher ensembles (PATE) [68]. This framework allows quantifying and binding the privacy exposure in a distributed setting of individual classifiers. One concept of quantifying privacy exposure is known as differential privacy (DP). If it can be proved that the revealed information is sufficiently hidden behind a mechanism that does not facilitate extracting the private information, this mechanism gives a privacy guarantee. The *PATE* framework provides a practical approach to use the gathered data contaminated with private information while bounding the privacy cost-effectively.
This work aims to evaluate if the classification model can use the *PATE* framework for large-scale classification of audio data. Visual data is out of scope here, while adaptions can quickly transfer the results to this medium. For this, an audio classifier is built on a publicly available audio data collection to demonstrate the applicability of the *PATE* framework for this use case.

1.2 Research objective

Building a classifier based on the private aggregation of teacher ensembles (*PATE*) framework is already extensively evaluated in the respective papers [68] and [67] from Papernot et al.. Other articles picked up their notable results regarding privacy-preserving classifiers. So far, there are no results for a multi-label audio-classifier with the *PATE* framework. In the original version,

PATE supports only multi-class classification. Sound snippets for the city and environmental sound naturally consist of more than one label of sound classes. The privacy analysis must address the increased information for the data labels within the *PATE* framework and its privacy analysis. In addition, audio data is far more complex than the evaluated data sets so far. As the information is split among the individual teachers for the *PATE* approach, the ensemble must train the individual classifiers with only a small data set.
Both aspects define the challenge to build an audio classifier for the city and environmental sound under consideration of the *PATE* framework. It should be evaluated how well a privacy-preserving audio classifier performs compared to a non-private model.

1.3 Organization

The work is organized into a theoretical section and the practical application for the defined use case. The theoretical part starts with information on the classification of audio data in the chapter 2. Afterward, a brief overview of privacy in machine learning is presented, and the original *PATE* framework is introduced. Related work and the placement to other research are discussed in the chapter3. The valuable part and own work begin with explaining the adaptation of the *PATE* framework and the application as an audio classifier for the *Audio Set* in the chapter 4. This section further discusses the multi-label privacy analysis. In the evaluation chapter,5 several experiments are conducted, and the resulting models are evaluated along with the steps in the *PATE* framework. The results are discussed with the last step of training a student model. The thesis closes with an overall discussion and conclusion in the chapter 6.

2 Theoretical background

The following chapter gives the reader the necessary background for the central part of the thesis. At first, the section 2.1 imparts general machine learning and audio classification knowledge. The section 2.2 continues with the information on privacy and its relevance for machine learning. Here the foundational models of differential privacy and Rényi differential privacy are introduced. The last part of the theoretical background familiarizes the reader with the *PATE* framework of Papernot et al. [68], [67].

2.1 Classification of audio data by machine learning

In machine learning for a given data set $\mathcal{D} = (x_1, \ldots, x_n) \in \mathbb{R}^{n \times r}$ of training samples a model $\theta \in \mathbb{R}^d$ is learned that minimizes some training loss. The loss is measured by a function $l : \mathbb{R}^d \times \mathbb{R}^r \to \mathbb{R}$. During the training process, a model is searched, that fits best the below condition:

$$\theta^* \in \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} l(\theta, x_i) \tag{2.1}$$

A standard model for all training samples should be found to minimize the aggregated loss. stochastic gradient descent (SGD), backpropagation for deep models, and many more techniques provide tool-sets to perform such tasks. Machine Learning can be mainly categorized into supervised or unsupervised learning. Labeled input data for a given feature vector is the base for supervised learning. Output values corresponding to the labels can be class labels (classification) or continuous values (regression). For unsupervised learning, no pre-labeled data is required. The most common application is clustering. Besides those two standard categories, mixtures of supervised and unsupervised learning exist and additionally reinforcement learning. The remaining thesis focuses on a supervised classification problem.

A training loop passes the samples through a machine learning model for classification. The mission of audio pattern recognition can be separated into audio tagging, acoustic scene classification, music classification, speech or natural language processing, and sound event detection. In the Audio Set case, this means assigning labels of sound events to the data. For all those different tasks, different techniques are more suitable to solve the task. The actual audio data or already extracted features usually are the input to those algorithms. Most audio pre-processing is based on some transfer to a spectrogram which can be seen as an image representing the sound snippet. For this kind of audio classification state of the art techniques of image processing can be utilized. Hence data scientists can apply common problems like edge, pattern, and object detection to audio classification.

There are former classification models that utilize manually-designed features audio-energy, zero-crossing rate, and a lot more features in the time or frequency domain. There are various applications of different machine learning models like hidden Markov models or support vector machines that this thesis will not discuss further here. As a state-of-the-art technique, CNN, are used for audio classifications. They are currently the best way to detect simple and highly complex patterns in the audio data [39].

For simplicity, the well-known models VGG and Resnets [38], [79] are used in the work. Both net architectures propose an building blocks for very deep networks. By those building blocks, they allow to create very deep networks without defining each layer individually. VGG is rather classical and consists of blocks of convolution and max-pooling layers. The numbers in VGG16 and VGG19 flavors denote the number of hidden layers.
The vanishing gradient problem describes an issue with such deep neural networks. In a training loop the backpropagation step pushes weight updates based on gradient calculation from the end of the network back through to fit the model to the task. The deeper the network becomes, the more the gradients are vanished by the weights it passed. Resnets address this problem by shortcuts between consecutive layers. During backward steps, they allow the gradient to pass more efficiently to the upper layers of the model. This design tweak regularly gives the Resnets a slight advantage over VGG architectures.
In each training cycle the loss function quantifies how good or bad a model's prediction is compared to the target output. This single value metric is used during the backpropagation step to adjust the inner parameters of the model. The binary cross-entropy (BCE) loss function is regularly used for multi-label classification. It allows to compare each class prediction individually and aggregate it to a single score.

Definition 1 (BCE loss)**.** *The BCE loss is defined below with t as binary target value and y as model output.*

$$Loss = -\frac{1}{|t|}\sum_{i=1}^{|t|} t_i \cdot \log y_i + (1 - t_i) \cdot log(1 - y_i) \tag{2.2}$$

Machine learning models are powerful tools for any classification task. They are the foundation for the classifier framework discussed in this thesis.

2.1.1 Nature of audio data

While visual and natural language processing tasks dominate the research topics around deep learning, regular audio analysis is a growing field. It includes processing digital signals, classifying or tagging sounds, and the artificial generation of audio. Audio processing has become publicly popular due to virtual assistants like Siri and Alexa. They support their owner with proprietary tools that can be accessed directly via speech and provide simple interfaces for developers to build extensions. While companies invested a lot of effort to improve those consumer-oriented recognition systems, these technologies now emerge in business applications.
Audio data is typically represented as amplitude over a time series. The amplitude describes the change in pressure around the recording unit that translated the initial sound into a signal. Our acoustic organ works similarly by translating changes in pressure into neural signals that are then further processed by the brain. Unless there are no further annotations available for the audio data, the time series is the only input for any audio processing. Those signals are continuous in the real world. For digital processing and storage space reduction, they are sampled to a discrete signal with a sampling rate that extracts data points according to this rate.
In addition to the time representation of audio signals, every signal can be decomposed into frequencies building up the audio. This correlates more to the human understanding of music

in which different frequencies create a specific sound. Humans hear the sound as pitch which is a subjective understanding of frequency. Sounds with higher frequencies have a higher pitch and vice versa. Lower frequencies have a smaller one. The *Audio Set* (explained in section 4.1) classification task is based on human-labeled sounds which is why the pre-processing should address the sounds similar to a human listener. The human acoustic organ is more sensitive to lower frequencies than higher ones. Systems can utilize the Fourier transformation to switch from time to frequency domain. According to change or other characteristics, there are hundreds of classical techniques to transform and describe the audio signal in the time or frequency domain. They got developed mainly for artificial music modification or simple system applications.
Nowadays, deep neural networks are used for audio-processing tasks. Even if it is possible to feed audio files directly to a deep learning network, it is uncommon. This would result in a one-dimensional array with the length of time multiplied by the sampling rate with the actual value. Representations in the frequency domain work much better for machine learning tasks. Hence the audio samples are decomposed into their frequencies within a specific time frame of the sample. The resulting representation is named spectrogram. It illustrates which frequencies appear with which intensity in the audio snippets over time. A nearly logarithmic scale in the frequency domain best represents the human understanding of sound.
The Mel-scale is the corresponding technique in audio-processing. It transforms the regular

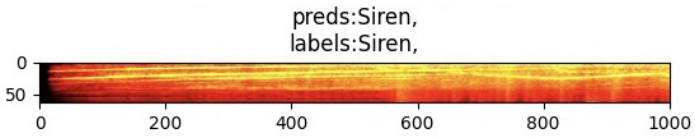


Figure 2.1: log-Mel diagram example for a “siren”.

spectrogram into a Mel spectrogram. Figure 2.1 shows an example of such an spectrogram. To reduce the range of frequencies and with it the dimensionality of the machine learning task, the Mel spectrogram is passed into filter windows. Such filter banks on the Mel scale have the characteristic that they become wider for higher frequencies which again correlates to the human understanding. Those filter banks are Mel bins representing the weighted amplitude of a particular range of frequencies. A human listener can hardly distinguish minor changes in frequency and the same understanding applies to the amplitudes in the time domain. To address this property, the decibel scale is used which model the increase of air-pressure exponentially. 10dB is 10 times louder than 0dB and 20dB are 100 times louder accordingly. Additionally the Mel bins are differently sized to redproduce the human understanding of sound. A Mel spectrogram that represents the Mel bins on the y-axis over the time on the x-axis and a decibel representation for the values in the diagram. In the shown example 64 Mel bins are shown over 1000 seconds. The color illustrates the intensity of the frequency at a specific time and the respective Mel bin.
The representation and pre-processing of audio data are essential for the application of machine learning. There exist many more tools that can extract features or representations from audio data. Mel spectrograms are famous for machine learning applications as it transfers the audio-processing task to a two-dimensional image classification task. This allows the application of all commonly known techniques for visual processing. As the focus of this work are not the audio

classification techniques themselves, the thesis will not consider more experimental techniques further.

2.1.2 Multi-class and multi-label classification

Classification tasks are the process of predicting classes for given data points. The goal of the task is to build a function f that maps input variable X to a discrete set of output variables y. Classifiers can be separated among the used techniques. The most commonly known is k-nearest neighbor, decision trees, naive Bayes, and artificial neural networks. Another way to categorize classification tasks is by the relation type between the input data and the labels. Machine learning theory can distinguish binary classification that allows only true or false, multi-class, and multi-label classification.

One of the most basic machine learning applications is multi-class classification. Formally multi-class prediction assigns a single label out of all possible classes to a sample. This assumes that the categories are not overlapping, and predictors can map every instance to precisely one of the classes. The output of a classification task is regularly a vector with the size of the available classes that represents the probability of the sample to be a class as the value. Revealing the full class vector might not be the best idea from a privacy standpoint. The aim of preserving information and binding the privacy budget for a classifier is to protect the data used for training. Leaking private data means how much personal information used for training becomes publicly available by using the classifier. Even if the information of probabilities for all classes might be helpful for training and optimization, they might not be necessary during inference. The first primitive idea to bound the privacy leakage for a multi-class classifier is to publish only the name of the highest probability class. Assuming there is a classifier detecting the language of a text. Several texts train it with the correct language attached. The consumer during inference is only interested in the resulting language and not outliers due to common words between the languages, for example. A single word that does not implicate the correct terminology but might be used in only one of the texts can lead to a high probability of incorrect language. This impacts the right result and indicates that the training loop used misleading text for training. Publishing the full class probability vector reveals more information about the inner workings of the classifier instead of just printing the resulting language.

In contrast, multi-label allows having multiple classes assigned to a sample. Hence a prediction of such a classifier creates a boolean vector of the form $\{0,1\}^c$. A threshold parameter is introduced to distinguish if a label must be assigned to a sample or not for a prediction. The formula implies that g becomes the sum of all individual results, and function h applies the noisy threshold to create the binary class vector. So far, there is no application of PATE to a multi-label scenario. Multi-label is a special form of multi-class classification. It also allows to assign multiple labels out of the label space to a sample. Hence, a sample gets a subset of labels assigned, ranging from zero to all possible. Both extreme cases do not make sense in most classification tasks, but there are exceptions. This concept is more usable for real-world data as most classification tasks are not binary and disjoint. The output of a multi-label classifier is again a vector of the size of the possible classes. A threshold distinguishes which classes are considered relevant instead of reducing this vector to the maximum value argument. Analogous to multi-class, as little as

the model should reveal possible information for the predictions for privacy reasons. Only the boolean label classifiers are published to preserve utility but protect the private information.

2.1.3 Performance metrics for classification tasks

Several metrics exist to describe the performance of a classifier. The overarching goal of every metric is to represent how good or bad the classifier performs the task. In the most simple scenario of a binary classification task, all data points have a deterministic outcome which can be positive or negative. Such a prediction of the classifier can be true or false depending if the assigned label is correct or not. The combination of the classification output and its correctness can be described in a confusion matrix.

There are some standard metrics often used to assess a classifier. Accuracy as essential metric describes the percentage of predictions which are correct. Precision expresses the classifier's rate of true positive classifications over all positive labeled samples. Whereas recall models the rate of samples classified positive by the algorithm out of all positive samples. Using those simple metrics helps to understand the performance of a classifier in a meaningful way. They are not described further as they should be well know to the reader.

Precision and recall always depend on each other. Hence they are often presented in a precision-recall diagram. It would be possible to tune recall that the classifier labels all samples with the class, which leads to lousy precision. To address this correlation, the F1 score computes the weighted average between accuracy and recalls. In contrast to simple averaging, the harmonic one penalizes extreme values in every direction.

As most classification tasks are not binary, an averaging method for the performance metrics is required. There exist two standard approaches to aggregate the classification performance per class. Micro and macro averaging are two significantly different approaches. While in the macro case, the metrics are calculated individually per class, the micro averaging calculates the performance for all classes together. The difference in those approaches is that macro treats all classes independently from the number of samples per class and its impact on the overall classification task. Micro averaging instead respects the number of samples per class. If the class imbalance is a known issue for the observed data, micro averaging represents the overall performance more precisely than the macro approach. On the contrary if the goal is to find a good classifier for all classes the micro aggregation distort the objective goal. Due to computational simplicity and independence from the evaluation data, macro averaging is used more often.

Definition 2 (Micro and macro averaged measures)**.** *In accordance to [81] the standard evaluation metrics Precisions (P), P and F1 are extended for multi-label classification. If not stated otherwise all mentions of these values refer to the macro averaged variant.*
Macro averaging measures:

$$\begin{aligned} P_{macro} &= \frac{1}{k}\sum_{i=1}^{k} P^{\lambda} \\ R_{macro} &= \frac{1}{k}\sum_{i=1}^{k} R^{\lambda} \\ F1_{macro} &= \frac{1}{k}\sum_{i=1}^{k} F1^{\lambda} \end{aligned} \tag{2.3}$$

Micro averaging measures:

$$\begin{aligned} P_{micro} &= \frac{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Z_i^j} \\ R_{micro} &= \frac{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j} \\ F1_{micro} &= \frac{2\sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j Z_i^j}{\sum_{j=1}^{k}\sum_{i=1}^{n} Z_i^j + \sum_{j=1}^{k}\sum_{i=1}^{n} Y_i^j} \end{aligned} \tag{2.4}$$

where

$$Y_i^{\lambda} = \begin{cases} 1, \textit{ if } x_i \textit{ actually belongs to class } \lambda \\ 0, \textit{ otherwise} \end{cases}$$

and

$$Z_i^{\lambda} = \begin{cases} 1, \textit{ if } x_i \textit{ is predicted for class } \lambda \\ 0, \textit{ otherwise} \end{cases}$$

Especially for multi-label scenarios, further important metrics exist. They allow reporting the performance independent from a specific threshold. While the class with the maximum score is predicted in a multi-class setting, multi-label requires an additional threshold that separates positive and negative predictions per class. Those other metrics are mean average precision (mAP) and area under the curve (AUC). Both are single value representations of the precision-recall diagrams.
average precision (AP) is the average sum of all precision values over all recall levels. This sum finds the area under the precision-recall curve for one specific class. As the whole precision-recall curve is observed, the result is independent of the threshold used to decide whether a class is a positive or negative representation. A skillful model has a precision-recall curve with a bump to the top-right, while a perfect model would be just a single point at $(1, 1)$. Hence an AP value of 1.0 would be an ideal classifier, and an AP close to 0.0 is a useless classifier. The threshold for the actual separation of positive and negative in a real-world scenario is then selected according to the best balance of precision and recall. Mean average precision (mAP) averages the AP scores for all classes.
ROC curves are the basis for the AUC metric. A ROC curve shows the true positive rate over the false positive rate for different threshold values. Hence it models false alarm over the hit

rate. This is again independent of a fixed classification threshold. Increasing the threshold by one step denotes on for a ROC curve that the next threshold includes one more sample on the false positive side. A bump on the curve to the top-left represents a skillful model, translating into a high true positive rate and a low false-positive rate. Hence a perfect expert model would be a single point at $(0, 1)$. Instead, a model without any skill is represented by a point at $(0.5, 0.5)$. If the model has no skill over a random mechanism at any threshold, this is shown as a diagonal to the top right from the origin. The ROC concept translates to a single value AUC by measuring the area under the ROC curve. An AUC of ≤ 0.5 describes a model without any skill, while an AUC value close to 1.0 describes a very skillful model.

2.2 Privacy

Privacy is a theoretical concept that allows an individual or a group to keep information about them hidden from others. Only information that they intentionally want to share does not conflict with their privacy. Protecting the privacy of an individual or a group means to ensure that information they are not willing to share remains private.
In the following sections the relation between privacy and machine learning is discussed and potential attack type to reveal private data are outlined. The key concepts of Differential Privacy (DP) and Rényi differential privacy (RDP) are introduced to the reader. Before moving over to the more sophisticated mechanism of providing privacy by the *PATE* framework in section 2.3 essential private mechanism are explained.

2.2.1 Privacy-preserving machine learning

Security and privacy guarantees for machine learning serve the more outstanding topic of data privacy [73] [42]. Data privacy moves into the focus with innovations around and by machine learning in all disciplines. Regulators introduced the GDPR objectives as "data protection by design and default." Privacy-preserving machine learning (PPML) describes the idea of applying this to machine learning. It spans training data privacy that protects the contributors against a malicious actor to model privacy that protects the model owner against stealing the model. Data processors must protect the input and output data for a model user during inference time against third parties eyes. Only the user who owns the data should see this data.
Separating the data and decentrally creating the model are the most apparent systematic approaches to privacy. Without moving the actual data or the model around, fewer risks exist for exposure. Federated learning is the set of techniques referred to here. Training the model is executed remotely on nodes where the actual data is stored to remain with the owner. A central handler must distribute only updated models (e.g., weights) over the network. The basic topology is flexible. The nodes can directly collaborate in a peer-to-peer infrastructure to share gained knowledge, or a significant party handles the communication. Even the data can be separated horizontally where the samples are split between nodes or vertically that demands the features to be divided between different parties. There are various successful implementations of federated learning already, like federated learning on Android devices [58].
All those efforts to secure data become meaningless without encryption. Data owners must apply cryptographic approaches to the transfer between nodes for model updates. But data modelers

can use cryptography within the actual training and model itself [40]. Homomorphic encryption is a scheme that allows direct computation of encrypted data. Most approaches are limited to base operations as addition, multiplication, or comparison, which developers can use to build up more complex functions. Standard encryption algorithms do not qualify to conform with these limitations. Instead, algorithms must use extraordinary mathematical transformations that show to be compute-intensive. Machine learning as a service is the most practical application for homomorphic encryption [31]. The service provider receives the data encrypted and does not need to decrypt it because he can perform the computation directly on the transformed information. Sharing data across multiple computation parties without them being able to restore the entire original data can be secured by secret sharing [55]. For this method, every party holds a private share of the data processed from all others. Only those with a threshold of all claims can aggregate and reconstruct the independently processed secret information. It is essential here that the computing parties do not collude. Various protocols exist for secret sharing with computation or other input parties that try to avoid such situations.
Next to encryption approaches in the protocol or computing, hardware-side technologies exist as well. Secure hardware implementations provide secure processors and enclaves that allow mathematical guarantees on a lower level. Initially, this invention was made to secure the execution of critical machine learning techniques and utilize them for further tasks such as machine learning. With even edge devices as smartphones containing such chips, this technology becomes more prevalent. Each data owner establishes independently a secure connection to the secure enclave where code and data are stored. Within the enclave, the processor can verify the integrity of the machine learning code and data uploaded using secure connections.
Perturbation approaches add fine-grained random noise to the machine learning task. Differential private mechanisms are well-known approaches that formalize noise to achieve some privacy guarantee. Any noise mechanism can be categorized by the machine learning task where the noise is applied. Datastores can directly add noise to the inputs. Even if the actual processing of the data with a non-private algorithm, the output is still private due to post-processing guarantees. If the data owners apply random noise by themselves, this corresponds to local differential privacy. Algorithm perturbation injects noise into the inner layers of an algorithm. For machine learning tasks, modelers can use similar techniques to prevent overfitting. Abadi et al. presented an approach to integrate DP into deep learning models [1]. The outputs can be blurred with noise, but this typically has a higher impact on the model's utility. The exponential accomplishes a better trade-off with a utility function that does not impurify the result and instead translates it to another scale that keeps all the information.

2.2.2 Threats

All machine learning algorithms have in common that they require big data sets to be trained. Machine learning is mainly applied to technical, medical, or monetary-driven purposes, so they process sensitive data. Especially the training data sets are either to be protected because of their inherent sensitivity (e.g., person data, disease data) or due to the cost-intensive creation process of the data set (e.g., manual labeling).
The personas involved in this process are the input party (data owner or contributors), the computation party, and the resulting model owning party [7]. Data samples are handed over

to the computing party that creates models out of the data. The result model party can then utilize the models for their purposes. All parties have a clear interest in keeping their data private. The individual contributors want to keep their information and characteristics not open to the world. In the past, computing and model-owning parties were in most cases the same. With upcoming machine learning as a service, it tends to becomes more and more separated. Respectively protecting the knowledge of how the data scientist built the model is in the interest of the computing party. As building up an excellent real-life model requires effort and money, an owner might not want to give it away at all.
The biggest threat remains that the private data are leaked in clear. Raw data is typically handed over to the computing party for the training process. Storing this data encrypted and not anonymized proposes a risk that insiders or outsiders access this data.

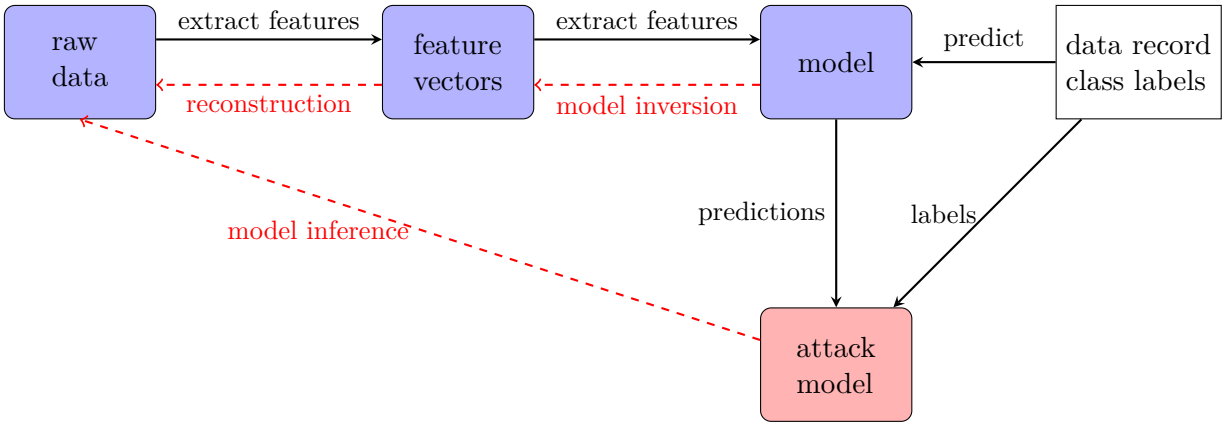


Figure 2.2: Visualized threats to private data in machine learning flow (in reference to [73]).

For already built models, three standard types of attacks exist [73]. A reconstruction attack aims to rebuild the raw input data based on feature vectors. Some models store such raw vectors in their inner layers and parameters. The goal for an adversary is to extract single properties or even complete data samples from the model. This type of attack requires white-box access to the model itself. Assuming a k-nearest neighbors (kNN) model that unveils the typical characteristics of a class in their classification approach is a simple example of such an attack. More complex would be the extraction of biometric data from a model used for authentication. Protection against such attacks is only possible by hiding the model itself from the result party by encryption or using models that do not store features inherently. There should be a further precaution against model inversion attacks to prevent the reconstruction of feature vectors.
In a model inversion attack, the adversary aims to retrieve some information about the inner working of the model. Here the adversary is limited to black-box access as the model does not store feature vectors directly, or only the results for queries to the model are available. The goal of such an attacker can range from reconstructing the whole model, over gaining knowledge about particular properties, or retrieving individual feature vectors in scenarios where the outputs of a model overlap with the input data it might even be possible to retrieve the actual input. Fredrikson et al. demonstrated an illustrative example of reconstructing input face images from a Generative Adversarial Networks (GAN) that should create artificial images [28]. A quota system can limit the attack surfaces by restricting the black-box access to the model. Limiting

the number of allowed queries or the information revealed by the answers are apparent techniques to defend against model inversion attacks. Instead of handing over the exact outputs of the model, it is sufficient for most applications to use rounded values or just the class labels without their probability.
Membership inference attacks reveal if a developer used a sample to train the model. Based on the knowledge of sample data, the real assigned labels, and the outputs of the attacked model, a new attack model can be built that predicts if the sample data is in the training set. Even the information if somebody is in some database can be sensitive. A person might not want to release the information about participation in a clinical or disease study. The team around Shokri et al. investigated such kinds of attacks [78]. They attacked several models with similar characteristics and distributions. One of them is trained on original data, but the shadow data sets are based on real noisy or synthesized data. Using only black-box attacks, they effectively predicted the actual data set and whether a data modeler used a particular sample to train this data set. With those results, they proposed and evaluated mitigation strategies for this issue. Again the most effective procedures are to restrict the information of the outputs. Besides this, regularization of the loss function during the training proved to be effective against membership inference attacks. This technique is usually applied to prevent overfitting, which is a shared goal between attack prevention and the model's accuracy. Both schemes can only be used to the extent that they disturb the models' utility only negligible.

2.2.3 Differential privacy

Differential privacy (DP) is a framework to publicly release the information of a data set while preserving the privacy of a single contribution. The patterns and characteristics of groups are valuable information to be removed, while mechanisms must protect the individual data point against adversaries. A trusted curator has access to a set of sensitive data D. It contains n records of a data universe X where each record holds sensitive data. This data set is protected from any untrusted data analyst for direct access. They query the data by asking the curator who answers by a randomized mechanism M. The desired level of privacy is achieved by adding perturbation to the result before providing the answer. DP aims to grant a privacy guarantee for the leakage of a single entry into a statistical database and identifying an individual with this information.
The idea here is that the effect of an arbitrary single substitution or the availability of a sample has a limited impact on the overall data set. With such a mechanism, a query can not infer much about any individual or its presence in the database. This guarantees privacy for people, companies, or other parties contributing to a database. For example, differential privacy is used by government agencies publishing demographic information, companies analyzing customer behavior, or scientific surveys trying to find patterns in personal data.
Cynthia Dwork et al. laid down the algorithmic foundation of Differential Privacy [22]. On this basis, many tools and techniques have been built to address privacy concerns in today's life. The definitions are taken over partially directly from the work of Dwork [22], from the papers of Papernot et al. [68], [67] and from Mironov [59].
A data set can be considered as a collection of records from a domain $\mathcal{X}$. The dimension of this space determines the information of a single document. Each data set x can be represented with a

histogram over $\mathcal{X}$ that depicts the frequency of a single data point for each dimension. Statistical queries are performed against the data set x. Differential privacy aims not to compromise the privacy of individuals who provided their data to this data set.

Definition 3 (Distance between data sets)**.** *Given data sets $D, D' \in \mathcal{X}^n$, the distance $d(D, D') = \|D - D'\|_1$ between them, can be modeled as the l_1 norm.*

It describes how many records differ between the databases. If $\|D - D'\|_1 \leq 1$ then the data sets are neighboring data sets. The effort to transform one data set into the other is to add or remove one record. Utilizing the norm distance fits well for theoretical application, but algorithms must use advanced methods to quantify the distance between two data sets in more complex scenarios. A random algorithm $\mathcal{M}$ reads the data set as input with optionally auxiliary data and returns some result. This maps the domain $\mathcal{X}$ of the data set to an output distribution $\mathcal{W} = range(\mathcal{M})$. Differential privacy guarantees that the probability of any outcome for such a mechanism is bound for a single individual. Any pair of neighboring data sets have a similar output distribution.

Definition 4 (ϵ-Differential Privacy)**.** *A randomized algorithm $\mathcal{M}$ is said to guarantee ϵ-differential privacy if for any neighboring data set D and D' and for any potential outcome $w \subseteq \mathcal{W}$ the following constraint is valid.*

$$P[\mathcal{M}(D) = w] \leq e^{\epsilon} \cdot P[\mathcal{M}(D') = w] \tag{2.5}$$

Where the probability is concerning the randomness in the algorithm $\mathcal{M}$.

The parameter ϵ in this equation refers to exposure or *privacy budget* risk. A small value for ϵ implies a lower privacy risk. ϵ-differential privacy is also referred to as pure differential privacy. Informally this technique can describe it as bounding a shift in the output distribution of a randomized algorithm that a minor change in the inputs can cause. ϵ acts as multiplicative upper bound for the worst-case change in the density of the distribution. It isn't easy to give such a strong guarantee in practical applications. Hence the (ϵ, δ)-differential privacy describes a relaxation.

Definition 5 ((ϵ, δ)-Differential privacy)**.** *A randomized algorithm $\mathcal{M}$ is said to guarantee (ϵ, δ)-differential privacy if for any neighboring data set D and D' and for any potential outcome $w \subseteq \mathcal{W}$ the following constraint is true*

$$P[\mathcal{M}(D) = w] \leq e^{\epsilon} \cdot P[\mathcal{M}(D') = w] + \delta \tag{2.6}$$

Where the probability is concerning the randomness in the algorithm $\mathcal{M}$.

Intuitively privacy is here described by ϵ and δ together. The new parameter δ quantifies the probability of failure that the equation does not hold for the ratio of probabilities. Bringing in a possibility of failure allows a relaxation of the privacy budget. If $\delta = 0$ then (ϵ, δ)-differential privacy simplifies to pure differential privacy.
Values of δ are typically less than the inverse of any polynomial in the size of the data set. Any larger δ is dangerous as this allows the full release of a subset of data points. For a fraction of records within the probability, there would be no guarantee against exposure.

To solve more complex problems, real-world tasks chain, and aggregate mechanisms. For a privacy analysis, a developer can disassemble them into building blocks. Differential privacy supports this by the below-described properties.
Mechanisms on a data set can be composed parallel or sequentially. Multiple disjoint subsets of the same data set are processed in parallel using different tools. The overall privacy budget corresponds to the worst sub-mechanism applied to any subgroup. In comparison, sequential composition refers to applying mechanisms on the same data set.

Theorem 1 (Parallel composition)**.** *For $i \in [k]$, let $\mathcal{M}_i$ be an ϵ_i-differentially private algorithm. Given a data set D, let $\{D^{(1)}, \ldots, D^{(k)})\}$ be k disjoint subsets of D. Let $\mathcal{M}$ be defined as $\mathcal{M} = (\mathcal{M}_1(D_1), \mathcal{M}_2(D_2), \ldots, \mathcal{M}_k(D_k))$. Then $\mathcal{M}$ is $\max_{i\in[k]} \epsilon_i$-differentially private.*

In a sequential mechanism $\mathcal{M}$ multiple subroutines $(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ are applied in sequence to the same data set. If the subsequent execution can utilize the outputs of the primary mechanism(s), it is considered adaptively.

Theorem 2 (Basic sequential composition)**.** *If a mechanism $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ such thta for any $i \in [k]$, $\mathcal{M}_i$ guarantees (ϵ_i, δ_i)-differential privacy, then $\mathcal{M}$ guarantees $\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i)$-differential privacy.*

Performing several sensitive queries against the same data set degrades privacy. Intuitively this corresponds to continuously asking more questions on a single record in the database. This reveals a more reliable answer with every new query than the actual accurate data. As this significantly limits the number of queries that a user can perform against a data set, keeping a defined privacy guarantee, there is a relaxation described in [23]. Instead of elemental linear composition, the privacy guarantees are combined sublinear with a trade-off in the probability of failure.

Theorem 3 (Advanced sequential composition)**.** *For all $\epsilon, \delta, \delta' \geq 0$, the class of (ϵ, δ)-differentially private algorithms satifies $(\epsilon', k + \delta + \delta')$ differential privacy under k-fold adaptive composition for*

$$\epsilon' = \sqrt{2k \log(1/\delta')} \cdot \epsilon + k\epsilon(e^{\epsilon} - 1) \tag{2.7}$$

A k-fold composition means applying k algorithms to the same data set, allowing adaptive changes for the queries to adversary permits to take the outputs of the k-th algorithm as inputs to the $k+1$-th algorithm. The advantage to the elemental composition is a growth of ϵ by $\sqrt{k}$ instead of linearity. In addition, all composed algorithms must have the same privacy (ϵ, δ). This is often not applicable for an intelligent adversary that adapts the queries to more revealing data point attributes.
Operations applied to the output of a differentially private algorithm do not affect the privacy risk post-treatment. Any post-processing of the data itself or combination with auxiliary data does not impact the privacy guarantees.

Theorem 4 (Closed under post-processing)**.** *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be an (ϵ, δ)-differentially private algorithm and $f : \mathcal{Y} \to \mathcal{Z}$ be an arbitrary function. Then $f \circ \mathcal{M} : \mathcal{X}^n \to \mathcal{Z}$ is (ϵ, δ)-differentially private.*

2.2.4 Rényi differential privacy

The RDP is another relaxation of ϵ-differential privacy [59]. Building on the work of Bun and Steinke [10], Mironov described a differential private framework based on the maximal statistical deviation in terms of Rényi Divergence and higher orders.
The concept of a privacy budget composition is the most relevant characteristic for the practical application of DP. For ϵ-DP, the advanced composition theorem adds up the privacy parameter. In case of the more tight (ϵ, δ)-DP there is a full class of $(\epsilon(\delta), \delta)$ parameters which lead to a combinatorial explosion for composition. RDP aims to provide a more effective and tight framework for composing mechanisms that satisfy several (ϵ, δ)-guarantees same time. RDP shares important properties like post-processing and composition with differential privacy. It can be easily converted to a class of $(\epsilon(\delta), \delta)$-guarantees.

Definition 6 (Rényi Divergence)**.** *The Rényi divergence of order λ between two probability distributions P and Q is defined as:*

$$D_\lambda(P\|Q) \widehat{=} \frac{1}{\lambda - 1} \log \mathbb{E}_{x\sim Q}[(P(x)/Q(x))^\lambda] = \frac{1}{\lambda - 1} \log \mathbb{E}_{x\sim P}[(P(x)/Q/(x)^{\lambda-1}] \tag{2.8}$$

Definition 7 (Rényi Differential Privacy)**.** *A randomized mechanism $\mathcal{M}$ is said to guarantee (λ, ϵ)-RDP with $\lambda \geq 1$ if for any neighboring data set D and D',*

$$D_\lambda(\mathcal{M}(D)\|\mathcal{M}(D')) = \frac{1}{\lambda - 1} \log \mathbb{E}_{x\sim\mathcal{M}(D)} \left[\left(\frac{P(\mathcal{M}(D) = x)}{P(\mathcal{M}(D') = x)} \right)^{\lambda-1} \right] \leq \epsilon. \tag{2.9}$$

RDP generalizes the pure differential privacy in the sense that ϵ-DP is equivalent to (∞, ϵ)-RDP. Temironov2017renyi the basic properties for composition and application to the Gaussian mechanism are proofed.

Theorem 5 (RDP composition)**.** *If a mechnism $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ such that for any $i \in [k]$, $\mathcal{M}_i$ guarantees (λ, ϵ_i)-RDP, then $\mathcal{M}$ guarantees $\left(\lambda, \sum_{i=1}^k \epsilon_i\right)$-RDP.*

RDP allows a much clearer description of complex mechanisms with the above-simplified composition. In addition, RDP captures the noise generated by the Gaussian distribution more accurately. Both characteristics together promote the usage of RDP for the Gaussian mechanism.

Theorem 6 (RDP of the Gaussian mechanism)**.** *Given a data set $D \in \mathcal{X}^n$ and a query function $f : \mathcal{X}^n \to \mathbb{R}$ and GS(f) bet the global sensitivity of the function, then the Gaussian Mechanism defined as $f(D) + \mathcal{N}(0, \sigma)$ guarantees $(\lambda, \frac{\lambda \cdot GS(f)^2}{2\sigma^2})$-RDP for all $\lambda \geq 1$.*

RDP can be easily transferred to (ϵ, δ)-DP to allow comparison with other algorithms and transfer other properties.

Theorem 7 (From RDP to (ϵ, δ)-DP)**.** *If $\mathcal{M}$ satifies (λ, ϵ)-RDP is also satisfies $\left(\epsilon + \frac{\log(1/\delta)}{\lambda - 1}, \delta\right)$ for any $0 < \delta < 1$.*

For λ the full class of $(\lambda, \epsilon(\lambda))$ are tracked for any $\lambda > 1$. This can be effectively converted to (ϵ, δ)-DP on a desired level of $\delta > 0$ by:

$$\epsilon = \min_{\alpha > 1} \epsilon(\lambda) + \frac{\log(1/\delta)}{\lambda - 1} \tag{2.10}$$

2.2.5 Private mechanisms

The essential tool to build a differentially private algorithm is to add noise to a query on a data set. This noise perturbs the actual result to apply carefully scaled randomness to the outputs. The art lies in scaling the noise concerning privacy but without destroying the usability of the algorithm. Two commonly used mechanisms are the Laplace Mechanism [24] and the Gaussian Mechanism [22]. The amount of required noise to achieve differential privacy is proportional to the sensitivity of the query function. The global sensitivity of a query function can be described as the amount of change if a single entry of the data set changes.

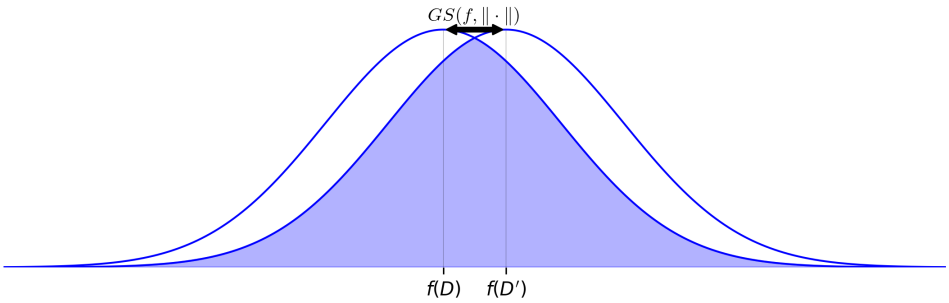


Figure 2.3: Output distributions for a function on neighboring data sets.

Definition 8 (Global sensitivity)**.** *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a query function and $\| \cdot \|$ be a distance metric. The global sensitivity of f with respect to $\| \cdot \|$ is defined as*

$$GS(f, \| \cdot \|) = \max_{d(D,D')=1} \|f(D) - f(D')\| \tag{2.11}$$

The sensitivity heavily depends on the type of query and the size of the data set. In a random data set representing university students, the query for the subject of study might be less unveiling than the query for a matriculation number. If a particular topic of study only has a handful of students, it also has a higher sensitivity.

Theorem 8 (Laplace Mechanism)**.** *Let Lap(β) be a random variable of a Laplace distribution Z with mean 0 and scale parameter β. Given a data set D and a mechanism that maps the data set to a result space $f : \mathcal{X}^n \to \mathbb{R}^d$. Concerning the privacy parameter ϵ and the L1-Norm used for global sensitivity, the Laplace Mechanism returns.*

$$f(D) + (Z_1, \ldots, Z_d)), \text{ with } Z_i \text{ drawn from } Lap\left(\frac{GS(f, \| \cdot \|_1)}{\epsilon}\right) \tag{2.12}$$

Past work can prove that the Laplace Mechanism provided ϵ-differential privacy [24]. The algorithm can use Gaussian noise to quickly achieve an (ϵ, δ)-differential private algorithm. Instead of Laplacian noise, Gaussian noise is used.

Theorem 9. *Let $\mathcal{N}(0, \sigma^2)$) be a random variable of a normal distribution Z with mean 0 and scale parameter σ^2. Given a data set D and a mechanism that maps the data set to a result space $f : \mathcal{X}^n \to \mathbb{R}^d$. Concerning the privacy parameters (ϵ, δ) and the L2-Norm used for global sensitivity, the Gaussian Mechanism returns*

$$f(D) + (Z_1, \ldots, Z_d)), \text{ with } Z_i \text{ drawn from } \mathcal{N}(0, \sigma^2) \text{ where } \sigma \geq \left(\frac{\sqrt{1 \log (1.25/\delta)} GS(f, \|\cdot\|_2)}{\epsilon} \right) \tag{2.13}$$

This mechanism guarantees (ϵ, δ)-differential privacy for any $\epsilon, \delta \in (0, 1)$. For proof see [22]. In figure 2.3 a basic example is illustrated. The random algorithm f is applied to two neighboring data sets, D, and D', and their output distributions are plotted. As the applied noise compensates for the global sensitivity, the two distributions become indistinguishable.

2.3 Private Aggregation of Teacher Ensemble (PATE)

Private Aggregation of Teacher Ensembles (PATE) refers to a framework introduced by Nicholas Papernot et al. for distributed and private learning [68], [67]. It builds on the "Bagging Predictors" idea of Breimann [8]. Bagging predictors include splitting training data into multiple subsets to train independent predictors. Aggregating them later can improve the accuracy of machine learning applications compared to a single model. The knowledge of numerous predictors becomes aggregated and transferred to a new model that profits from the diverse models and a proper aggregation strategy.

Aggregation approaches with distinct models provide good results for the following reasons. Machine learning techniques typically allow multiple well-fitting representations of the training data for an optimization goal like accuracy. This holds especially if the samples are limited and no perfect fit can be achieved. Instead of relying on a single representation for the space, ensembles profit from reducing the statistical error by aggregation [20]. Most machine learning algorithms try to find some extreme that represents the data best. Even with unlimited samples, an algorithm might find a local minimum for the loss function instead of the global one. An ensemble allows the training from different starting positions and separate training sets, resulting in a better approximation for the overall data set.

For *PATE*, a teacher ensemble is trained on disjoint subsets of the private training data. Each teacher has a sub-set of the data and trains a machine learning model. The subsets can be created randomly or use existing, natural separations. For example, models on medical data can be trained directly by the hospital itself instead of a central curator that has access to all data. While not all models do not have to use the same techniques or model types, they must share the common goal and function to serve the utility of the ensemble.

New non-private auxiliary data of the same domain is used for the student model training. The ensemble predicts and aggregates the samples for these data samples. Based on the aggregated predictions of the teacher ensemble, a student model is trained. Only through the aggregated

predictions of the teacher ensembles some private data becomes unveiled to the student. Only the ensemble has access to the extracted knowledge of the models trained on the private data. Intuitively this technique avoids that the student model depends on a single data point of the private data. Even with white-box access to the student model, it is impossible to directly retrieve private data from the setup. The information has never been shown directly to the student and the teacher models will not be publicized.
Privacy seems to be intuitively granted for a model trained only on quorum votes of a teacher ensemble. Nevertheless, privacy is always hard to reason about as each private sample can impact the student model significantly. Even if they are only presented to the teacher the ensemble can leak private information to the student. Differential privacy is used to formally describe and quantify the influence of the teacher ensemble votes on the student model. The aggregation mechanism exposes the private data to the student training. Hence applying well-calibrated noise to the aggregation step can constitute a privacy guarantee. Due to post-processing guarantees, the student model itself becomes inherently private. No further private information can be accessed after the student model is trained.
The first paper for "Semi-supervised knowledge transfer for deep learning from private training data" was published in 2016. It outlines the approach and the framework [68]. For privacy analysis, the authors use the *Moments Accountant* from Abadi that monitors the consumed privacy budget in a data-dependent approach directly in the algorithm. They introduce PATE-G, a GAN for the student training and apply their methods to essential data sets as MNIST [16] and SVHN [63]. In the updated paper "Scalable private learning with PATE," they seek to apply and evaluate their framework to more extensive, real-world data sets [67]. To accomplish this, they move away from the GAN approach and replace the *Moments Accountant* by applying RDP. With the RDP privacy framework, they develop dedicated proofs to address the release of privacy by the ensemble as accurately as possible.
The following discussions mainly refer to the updated version of *PATE* and leave the *Moments Accountant* and GANs out of scope.

2.3.1 PATE framework

The *PATE* approach is a sequential flow illustrated in the image 2.4. A potential adversary can only access the student model and the public data right of the grey barrier. All other parts of the framework must not be published. The actual learning from the private training data happens hidden from any third party. This part is referred to as teacher ensemble and is further described in section 2.3.2. Through various steps, private information flows from the data on the left side to the student model on the right.
Due to the outlined possibilities and threats of white- and black-box attacks against the model in 2.2.2 privacy guarantees must address the worst-case scenario. It must be assumed that an attacker has unlimited access to the publicly reachable model and information. For *PATE* this means even the potential transfer of private information in the student model from the private teacher ensemble must be considered. The potential privacy leakage that must be quantified and bounded inherits from the knowledge transfer from the teacher ensemble to the student model. This is illustrated by the slight overlap of the prediction for the public data across the barrier.

A detailed explanation of privacy analysis can be found in sections 2.3.3 and 2.3.4.

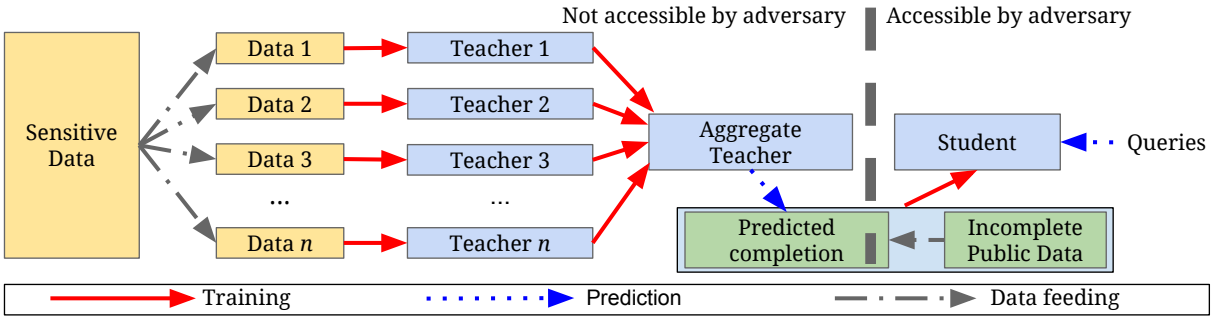


Figure 2.4: PATE framework overview [68]

The work on *PATE* mainly focuses on two key topics that significantly impact the utility and the privacy of the *PATE* framework. First is the degree of consensus for the predictions among the teachers for a non-private sample. The privacy cost for using the aggregated ensemble prediction is much smaller if more teachers agree, and the overall prediction does depend less on the single vote. Informally this can be described as the single teacher vote is hidden behind the ensemble. The intuition of *PATE* to address the first one is to build on strong consensus. Assuming that a group of people answer a yes and no question. If they agree, it is less likely that the overall answer depends on a single person. The impact if an individual contributed to the answer or not is limited. This is addressed in the *PATE* papers as data-dependent privacy analysis and below in section 2.3.5. Secondly, the number of queries required to the teacher ensemble to train the student model. Every quorum collected from the ensemble impose a privacy cost, which adds up to the spent privacy budget. Hence training the student with as few samples as possible leads to stricter privacy guarantees. Both topics will be reflected in more detail later.
On the contrary, if the answers are equally balanced, it is more complicated to say how the individual influences the answer. It further matters how the votes are released to the public. Does only the information, if yes or no wins, are published, or are the actual counts or percentages released? All this impacts the spent privacy budget for a query. The spent privacy budget adds up by the number of questions asked. Suppose multiple questions are answered, like how many people voted yes and have a name starting with "A" to train the student model. *PATE* seeks strategies to minimize the cost of a query as good as possible and limit the number of queries against the teacher ensemble to bound the privacy budget effectively.

2.3.2 Teacher ensemble training and aggregation

For the teacher ensemble individually teacher models are trained on a subset of the sensitive training data. All together, build the teacher ensemble. Instead of using the entire data all at once, each teacher trains on a disjoint part of the data. To accomplish this, the training set (X, Y) where X denotes the samples and Y the corresponding labels are partitioned into n disjoint sets (X_n, Y_n). Based on each subset i, a teacher model f_i is trained. It does not matter what type of model or technique is used to train the teachers. Even different models per teacher are possible as long as they are compatible with the format and dimension of the data.
The performance of an individual teacher model depends heavily on the subset of the training

data. If the size is unappropriated compared to the problem to solve or if it might be heavily imbalanced, this might impact the individual performance. For the *PATE* knowledge transfer to the student model, not the individual teacher matters, but the ensemble's performance. It might even be profitable for the overall result if some teachers specialize in particular tasks.
Understanding the whole set of teachers as an ensemble of size n opens up how to use it to create predictions for new samples. A sample is passed through all teacher models $f_i(x)$, and a single prediction is aggregated by their outputs. This approach provides privacy by hiding the individual sensitive information that a single teacher poses behind the ensemble aggregation. As mentioned earlier, this works best if most teachers agree. The strength of the privacy guarantee derives from the mechanism used for aggregation. As for most differential private mechanisms, noise is added. For *PATE* a random value is applied to the aggregation algorithm for teacher votes.

Definition 9. *The prediction for a sample x by a teacher ensemble of size n is defined by an aggregation function g for the individual votes $f_i(x)$ of the teachers and random noise $\mathcal{N}$ drawn from a distribution. The function h normalizes the noisy aggregated result of the ensemble to a prediction.*

$$f_{ensemble}(x) = h(g(f_0(x), f_1(x), \ldots, f_n(x)) + \mathcal{N}) \tag{2.14}$$

The work on *PATE* by Papernot et al. addresses multi-class prediction [68]. The formula simplified here as g represents the count histogram for the classifier. h is defined to retrieve the class with the maximum number of votes. This corresponds to plurality voting among the teachers. Every teacher has a vote assigned to the class with the highest probability. Hence the ensemble histogram functions g acts as a classifier itself.

2.3.3 PATE privacy analysis

The global sensitivity (GS) defined in section 2.2.5 quantifies the impact of a single data element on the predictor. This means it describes how much the presence or absence of a single training sample potentially changes the classifier's output. The Global sensitivity (GS) can be intuitively bounded for a multi-class classification scenario that only outputs the winning class. In the case of a single predictor, the presence of a sample during training can impact the model so that it flips from one class to another. In an extreme scenario, it flips the probabilities of the two classes from 0.0 to 1.0 and the other way around. Hence the upper bounded global sensitivity for multi-class classification is $GS(f_i, \|D - D'\|_1) = 2$. This holds similarly for the ensemble scenario. A sample is present or absent for exactly one teacher. Hence the impact of this sample is limited to exactly one teacher. However, each of the individual classifiers can flip the overall prediction by adding and removing their vote for the two histogram bins. The overall global sensitivity of a teacher ensemble is as well $GS(f_{ensemble}, \|D - D'\|_1) = 2$.
The privacy of the *PATE* framework derives from the noise applied to the aggregation mechanism of the teacher ensemble. It adds randomness to the aggregated result of the teacher ensemble to prevent that the private information of an individual teacher from heavily influencing the result. There is no noise applied directly to any individual classifier, but only to the aggregation. The overall task of the ensemble is to predict the class label for a sample. The aggregation mech-

anism is a randomized mechanism that can ensure privacy guarantees. For the below analysis, only Gaussian noise with the RDP framework is used as introduced in section 2.2.4. The first *PATE* paper [68] utilizes Laplacian noise and the *moment's accountant* approach to measure the data exposure, which got replaced by the RDP analysis that allows tighter results. Another advantage of RDP is that the utilized privacy budget per query to the teacher ensemble adds up in the RDP domain easily. This composition property is explained in section 2.2.4.
The overall intuition for privacy bounds is to model them as tight as possible. It is easy to describe a big upper bound with the same size as the algorithm could leak in the worst case. Below privacy, the analysis starts with an independent data view on the aggregation mechanism. Later in the next section 2.3.4 a tighter data-dependent analysis for the teacher ensemble is presented.
The PATE aggregation for a multi-class scenario gaussian noise max (GNMax) utilizes a histogram h of votes from the teachers. Each teacher has one vote representing the class with the highest likelihood of the classification algorithm. During the aggregation, Gaussian noise is added to the histogram. The algorithm outputs aggregated votes with Gaussian noise with mean 0 and variance σ^2.

$$\mathcal{M}_\sigma(x) \triangleq \arg\max_i \{h_i(x) + \mathcal{N}(o, \sigma^2)\} \tag{2.15}$$

The mechanism satisfies $(\lambda, \lambda/\sigma^2)$-RDP for all inputs and all moments $\lambda \geq 1$. This result can be directly derived from the structure of the aggregation. A teacher can impact two counts on the histogram. One is incremented, the other one is decremented. That corresponds to the global sensitivity of 2. As the GNMax is a Gaussian mechanism, it fulfills $(\lambda, \lambda/\sigma^2)$-RDP. This privacy analysis is referred to as data-independent privacy analysis.
With the addition of noise, the PATE framework aims to give a privacy guarantee. One of the main concepts of differential privacy is to bound the difference in the result of an algorithm based on the question of whether a single sensitive data item contributed to it or not. Only one teacher has seen this sensitive data within the teacher ensemble as they operate on disjoint data sets. Applying noise to the teacher ensemble aggregation intends to bound the impact of a single teacher to bound the privacy leakage. The noise induced to the aggregated teacher predictions impact the aggregated prediction of the ensemble. If the noise is too big or even exceeds half of the teacher votes, the aggregation becomes useless. The other way around, if it is too tiny, a single teacher can define the outcome of the ensemble.
Well-calibrated noise prevents an adversary from distinguishing if a single teacher flips the aggregated prediction or the induced random noise. If the ensemble does not have a strong consensus, this might require more sampled noise. For example if half of the teachers vote for one class and the other half for the other, the expected value of the noise should be at least as significant as the impact of a teacher on the ensemble result. On the contrary, less noise must be injected if there is a strong consensus. Where all teachers independently agree to the same result it does not depend on any individual. This observation will be addressed as data-dependent privacy as it takes the votes of the ensemble into account.

2.3.4 Improved data-dependent privacy analysis

To enhance the data-independent analysis and provide tighter bounds, the properties of the predicted votes are used. The intuition behind the data-dependent analysis is that if the consensus between the teachers is high, the aggregated result depends less on the private inner structures of an individual teacher. In practice, the data-dependent bound can be worse than the data-independent bound. To address this in the privacy analysis, the minimum between those two bounds are considered for each sample.
Instead of treating each ensemble prediction similarly, the degree of consensus of the teachers becomes relevant for the privacy analysis. As outlined already, a strong consensus decreases the ensemble vote's dependence on the individual teacher. It allows smaller privacy bounds compared to the data-independent approach. If the consensus is not strong enough or does not exist at all, this leads to a wide data-dependent bound. This intuition is transferred to a probability $\widetilde{q}$ that approximates if the most common answer will not be output by the mechanism. An outcome i^* has the maximum support of the teacher ensemble. The probability that the mechanism does not output i^* can be described as $Pr[\mathcal{M}_\sigma(D) \neq i^*] \leq \widetilde{q}$.
For a multi-class scenario, an outcome i^* with the maximum support of the teacher ensemble means that the majority of votes are assigned to this class. The probability that this is not the case can be quantified by the sum of the probabilities that any other class wins over the most likely class. A class with the maximum assigned votes can be not the output if a competing class overcomes the difference of class votes with the added noise. The probability for a random variable of a normal distribution of exceeding a value can be calculated by the error function (corresponds to cumulative distribution function (CDF) of Gaussian distribution). The value is described by the difference between the class votes and the class with maximum number of votes. Twice the noise is used as variance. The doubled variance inherits from the fact that two counts can be changed by the applied Gaussian noise with σ-variance. For details and proof, please see the appendix of the *PATE* paper [67].

Theorem 10. *In a multi-class scenario for any class $i^* \in [m]$ the probability that a outcome will not be outputted by the mechanism can be described as $Pr[\mathcal{M}_\sigma(D) \neq i^*] \leq q(\bar{n})$. Where $q(\bar{n})$ is a function of the vote histogram $\bar{n} = (n_1, \ldots, n_m)$ with $q(\bar{n}) = \sum_{i \neq i^*} \frac{1}{2} \left(1 - erf(\frac{n_{i^*} - n_i}{2\sigma})\right)$.*

The theoretical work of the *PATE* paper lies in the usage of the above data-dependent analysis that result in RDP guarantees. The idea of the proof will be sketched here only. For details, please refer to the appendix of the original paper [67].

Theorem 11. *Let $\mathcal{M}$ be a randomized algorithm with (μ_1, ϵ_1)-RDP and (μ_2, ϵ_2)-RDP guarantees and suppose that there exists a likely outcome i^* given a data set D and a bound $\tilde{q} \leq 1$ such that $\tilde{q} \leq e^{(\mu_1 - 1)\epsilon_2} / \left(\frac{\mu_2}{\mu_1 - 1} \cdot \frac{\mu_2}{\mu_2 - 1}\right)^{\mu_2}$. Then, for any neighboring data set D and D', the following applies.*

$$D_\lambda(\mathcal{M}(D) || \mathcal{M}(D')) \leq \frac{1}{\lambda - 1} \log \left((1 - \tilde{q}) \cdot \boldsymbol{A}(\tilde{q}, \mu_2, \epsilon_2)^{\lambda - 1} + \tilde{q} \cdot \boldsymbol{B}(\tilde{q}, \mu_2, \epsilon_2)^{\lambda - 1}\right) \tag{2.16}$$

where $\boldsymbol{A}(\tilde{q}, \mu_2, \epsilon_2) \stackrel{\Delta}{=} (1 - \tilde{q}) / \left(1 - (\tilde{q} e^{\epsilon_2})^{\frac{\mu_2 - 1}{\mu_2}}\right)$ and $\boldsymbol{B}(\tilde{q}, \mu_2, \epsilon_2) \stackrel{\Delta}{=} e^{\epsilon_1} / \tilde{q}^{\frac{1}{\mu_1 - 1}}$.

The intuition of this approach here is that two higher-order data-independent RDP guarantees are translated to a lower order RDP guarantee using $\tilde{q}$. By definition, Rényi-Divergence uses

the expected values of the distributions. In the multi-class scenario, the possible outcomes are discrete. This can be separated into the sum of the likely event $(1 - \tilde{q})$ and all other events $\tilde{q}$. Hence the term $\boldsymbol{A}$ aims to improve the probabilities using inequalities and estimations based on the likely outcome q. Similarly, this applies to $\boldsymbol{B}$ for the unlikely outcomes. With this they introduce above inequation for the RDP of order λ as a function of $\tilde{q}, \mu_1, \mu_2, \epsilon_1$ and ϵ_2.
The actual proof is based on the idea that there is one likely outcome i^* and the corresponding probability $Pr[\mathcal{M}(D) \neq i^*]$ that the output of the algorithm will not be i^*. All remaining probabilities are described relative to this event. There are no assumptions about the number of classes or events. With translating the two possible outcomes to a higher order a tighter expected value can be formulated. Combining this with the probability $\tilde{q}$ a estimated upper bound for the real privacy cost of q can be calculated.

$$\mu_2 = \sigma \cdot \sqrt{\log(1/\tilde{q})}, \text{ and } \mu_1 = \mu_2 + 1 \tag{2.17}$$

For practical application optimal values for the data-independent orders, μ_1 and μ_2 must be found. The theorem 11 allows them to translate RDP bounds to a lower order λ data-dependent bound. The goal is to have the RDP bounds as tight as possible. Therefore the equation 2.16 must be minimized on μ_1 and μ_2. As they only appear on one side of the equation, they can be minimized individually. The approach for the local minimum for the GNMax aggregator with a variance of σ^2 can be found in the PATE paper [67]. For ϵ_1 and ϵ_2 the data-independent bounds are considered of theorem 10. The higher orders μ_1 and μ_2 for the data-dependent analysis are chose according to above remarks. It is possible to calculate a critical value of q for this parameter to decide if it is worth to perform a data-dependent analysis or if the data-independent will be tighter anyway. Only if the ensemble predictions have a strong consensus the data-dependent privacy analysis is tighter than the data-independent one. Obvious conditions to perform the data-dependent analysis are $q < 1$, $\mu_1 \geq \lambda$, $\mu_2 > 1$, and $q \leq e^{(\mu_2-1)\epsilon_2}/(\frac{\mu_1}{\mu_1-1} \cdot \frac{\mu_2}{\mu_2-1})^{\mu_2}$ as discussed in detail in the appendix of [67].

$$\beta_\sigma(q) \triangleq \min \left\{ \frac{1}{\lambda - 1} \log \left((1 - \tilde{q}) \cdot \boldsymbol{A}(\tilde{q}, \mu_2, \epsilon_2)^{\lambda-1} + \tilde{q} \cdot \boldsymbol{B}(\tilde{q}, \mu_2, \epsilon_2)^{\lambda-1} \right), (\epsilon, \lambda) \right\} \tag{2.18}$$

Combining all the previous statements creates the recipe for measuring the spent RDP privacy budget for a prediction by the teacher ensemble for a given data set D. This building plan applies for multi-class GNMax. The raw outputs of the teacher ensemble aggregation are used for the privacy analysis. The privacy cost must be calculated for each query to the teacher ensemble. Due to the properties of RDP, the individual privacy budgets is added in the RDP domain. The summed RDP cost can then be translated into ϵ, δ-differential privacy ((ϵ, δ)-DP) guarantees for a fixed λ.
For the final calculation of the privacy budget for a query to the ensemble, the minimum of the data-independent analysis of theorem 10 and the data-dependent analysis of the theorem 11is taken.
Overall the privacy cost depends on the applied noise σ and the predictions of the ensemble. Unfortunately the applied noise impacts also the prediction quality of the ensemble. The privacy cost is in a trade-off with the prediction capability of the ensemble. If the individual teachers can create a strong consensus due to highly capable classifier the data-dependent analysis allow much

tighter privacy guarantees. The worst-case privacy cost upper-bound is the data-independent analysis with the RDP framework. In the best case the data-dependent analysis results in a much tighter privacy guarantee for this sample.

2.3.5 Student model knowledge transfer

The teacher ensemble of *PATE* can answer queries with a calculated privacy guarantees. Using the ensemble during inference time improves utility and accuracy compared to using the knowledge of the individual teacher models. Besides this, the injected noise to address privacy can be calibrated according to the consensus of the individual models. The privacy budget disposed of for an answer mainly depends on the consensus among the teachers and the amount of noise injected.
However, there exist two limitations the teacher ensemble can not address itself. An additional privacy budget is spent with each new query to the teacher ensemble. The privacy budget composes according to the composition rule of RDP. Letting an adversary directly use the teacher ensemble to predict during inference time would lead to a steady increase of the privacy cost. If there is a hard limit on privacy, a new teacher ensemble must be trained every time it is reached. This requires new data that is usually not available. Directly publishing the teacher ensemble or individual classifier models would bring additional challenges. Releasing those models open up the surface for model inversion attacks. To ensure privacy, one trusted aggregating party must be acquired that controls the access to the models and sanitizes the predictions. All this seems to be impractical to guarantee privacy for the data.
To address those two drawbacks, the *PATE* framework trains a student model. The knowledge of the teacher ensemble is transferred in a privacy-preserving way to a student model. For this it utilizes unlabeled and publicly available data that was not used to train the teacher ensemble beforehand. Depending on the data domain, it is more or less complicated to acquire such privacy irrelevant data.
The private data samples that should be protected are never shown to the student directly. Hence, privacy leakage occurs only within the knowledge transfer mechanism from the ensemble to the student. While training the student model, the samples are passed to the teacher ensemble for labeling. Due to the noisy answers of the ensemble, the privacy cost of the transfer from the teachers to the student can be quantified. Even more critical, limiting the number of queries to the teacher ensemble allows achieving fixed privacy guarantees. They are independent of how many queries are raised to the student model later on. But any additional query to the teacher ensemble increases the spent privacy cost. Only the student model must be published for inference time. The overall privacy cost of the label assignments on the public data can be quantified.
This knowledge flow is briefly described in the illustration 2.5. The private information of 'Jane Smith' is available only in one of the teacher models. This private information is transferred to the inference time student model only by labeling the teacher ensemble's public data. The leakage of this private information will be quantified by analyzing the aggregation mechanism of the teacher ensemble. The dashed lines on the bar diagram are the added noise.

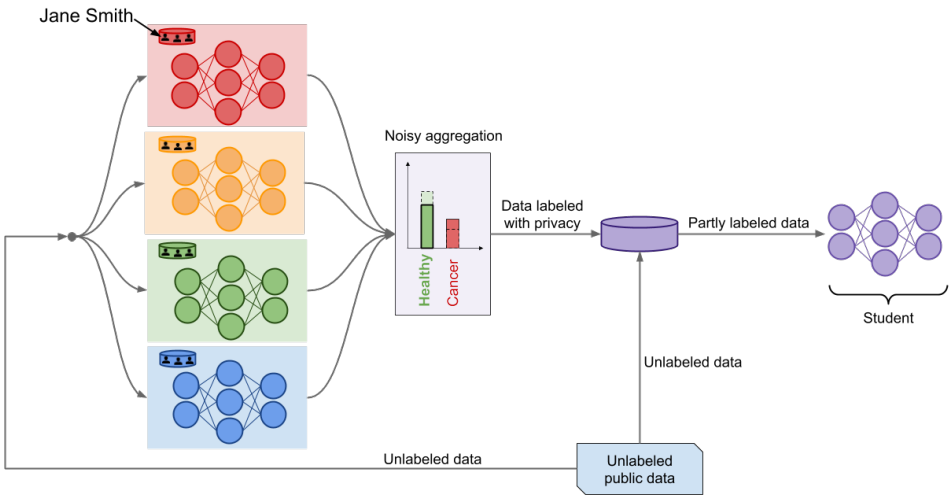


Figure 2.5: Private knowledge transfer to the student model [66].

There exist ideas for improvement for this basic approach by the authors of the papers. The intuition behind all approaches is to emphasize high consensus within the ensemble. This ensures that the information learned from the sample is helpful and allows tighter privacy guarantees as described in 2.3.3. The approaches range from distillation or model compression over active learning techniques to a GAN-based approach [68]. Most of the attempts have no sufficient balance between the increase in accuracy compared to the additional used privacy budget. Two promising approaches are outlined below.

2.3.6 Improved aggregation and knowledge transfer methods

Not every sample contributes similarly to the training of the student. Their class predictions can be either ambiguous or complicated to classify in general. Only binary labels are used for student training to limit the transferred private information to a minimum. Weak class votes that only have a slight advance above other competing classes do not contribute similarly to a crystal clear class assignment. The confidence aggregator makes use of this property [67]. Each query to the ensemble is analyzed for the confidence of the teachers. Samples that do not exceed the desired consensus are dropped. This technique contributes to the model's accuracy and helps achieve tighter privacy guarantees. The higher the confidence of the ensemble is, the lower the impact of an individual teacher on the overall prediction.
The confident GNMax aggregator is described below algorithm 1. A threshold th_c separates samples with a sufficient consensus from weak agreements between the teachers. For this step, the private data is utilized. To perform this comparison in a privacy-preserving way, Gaussian noise with the variance σ_1^2 is added to the check. For samples where the sample passes the threshold, the algorithm answers with a significantly smaller variance σ_2^2. Otherwise, no answer will be provided by the ensemble.
The confidence GNMax algorithm's privacy costs consists of the costs imposed for answering the threshold check and the sample prediction. If the actual query to the ensemble is not answered, the threshold check impacts the privacy budget regardless. The intuition here is that if queries with an extensive privacy cost will not be answered, this compensates for the additional privacy costs for the threshold check. As the threshold privacy cost is always consumed, the noise factor

Algorithm 1 Confident GNMax aggregator [67]

Input input x, threshold T, noise parameter σ_1 and σ_2

1: **if** $\max_j\{n_j(x)\} + \mathcal{N}(0, \sigma_1^2) \geq th_c$ **then**
2: **return** $\arg\max_j\{n_j(x) + \mathcal{N}(0, \sigma_2^2)\}$
3: **else**
4: **return** $\perp$

for σ_1^2 is often chosen higher than for σ_2^2. The threshold itself values higher than half of the number of teachers in the ensembles are chosen. Queries with more minor agreements between the teachers are likely not to be answered to avoid privacy cost-intensive operations.
The interactive mode of the aggregator aims to check if the sample contributes to the learning of the student model or if the student has already successfully predicted the correct labels. In contrast to the confident aggregator, this does not favor strong consensus within the ensemble but checks the consensus between the ensemble and the student. If the student predicts the labels successfully, then there is no need to spend again privacy budget for such a sample.
The interactive GNMax aggregator is described below algorithm 2. The maximum difference of class votes between the scaled student predictions and the teacher ensemble answers are compared in a privacy-preserving way. To make this mechanism, privacy-preserving noise is added to the difference. If the difference for this class exceeds the threshold th_i, then the student is considered to be not confident in predicting this query. In this case, the query are answered with the regular GNMax aggregation. If the difference threshold is not violated in a second step, the student prediction is checked for confidence. Only if the prediction by the student model exceeds another threshold th_{ic} does the student prediction become reinforced. For this, the answer of the student model is taken as labels for the new sample. There is no need to apply noise to this step as the student model contains only public data. If there is no high deviance between the student and the teacher ensemble, but the student is not confident, there is no reinforcement.

Algorithm 2 Interactive GNMax aggregator [67]

Input input x, student prediction p_j, threshold ensemble th_e, threshold student th_i, noise parameter σ_1 and σ_2 and the number M of teachers

1: **if** $\max_j\{n_j(x) - Mp_j(x)\} + \mathcal{N}(0, \sigma_1^2) \geq th_i$ **then**
2: **return** $\arg\max_j\{n_j(x) + \mathcal{N}(0, \sigma_2^2)\}$
3: **else if** $\max_j\{p_j(x)\} \geq th_i$ **then**
4: **return** $\arg\max_j\{p_j(x)\}$
5: **else**
6: **return** $\perp$

The consumed privacy consists of the noisy computation of the consensus check between the teacher and student predictions and the regular noisy answering of the prediction. Only in cases where the student prediction deviates from the ensemble prediction the query is answered by the teacher ensemble. If there is no need to correct the current behavior of the student, no additional privacy are consumed except for the comparison. As long as the student is confident enough, the prediction are reinforced without inspecting the teacher votes again and spending any further

privacy budget. Due to the ordering of the statement, it is prevented that incorrect predictions will be reinforced.
It does not make sense to start the training process with the interactive GNMax aggregator for training the student model initially. The student model can not confidently predict any sample without training upfront. Hence the student should be trained with the interactive mechanism only at a late stage in the training process.
The confidence GNMax aggregator and the interactive GNMax aggregator are powerful tools to reduce the privacy costs of a student model even further. They are practically applied to the experiment of Papernot et al. and have a significant share on the privacy results they achieve.

3 Related work

The below chapter outlines related research objectives and sorts the work into a field of study. For this, three different areas are considered. Differential privacy, in general, is followed by private learning from distributed data. The third section looks at other approaches to building an audio classifier and similar works on the *Audio Set* data[29].

3.1 Privately learning from data

While differential privacy and variants of it are the standards for guaranteeing privacy nowadays, there have been several approaches before. The *Tracker* mechanism of Denning et al. tries to formalize a way to track all previous queries to the database and prevent revealing data based on the current and all last queries [17]. Due to high dimensional data and the overall complexity of such tracking. Sweeny together with Samarati worked on minor tools for the private release of medical documents to third parties [84], [76]. Their work aggregated to the well-known *k-anonymity*, which aims to release data only if a person can not be distinguished from at least $k-1$ other data sets [83]. Information that does not fulfill this constraint must be generalized or suppressed. This shows that it is impossible to publish answers to queries on confidential data while restricting the reveal of private information. The mitigation approaches of *k-anonymity* are considered one of the first formal attempts to solve the privacy problem by adding noise to the results, which build the path to differential privacy. *k-anonymity* becomes problematic for high-dimensional data with thousands of attributes [4]. There possible outputs grow exponentially while *k-anonymity* requires to consider all possible outcomes.
Besides this, several attempts as *l-diversity* [53] and *t-closeness* [50] try to mitigate the shortcomings of *k-anonymity*. With the illustrative de-anonymization attack and formal descriptions of possible attacks against the Netflix price, data set Narayanan and Shmatikov demonstrated the weakness of *k-anonymity* [61].
There exist several ideas to implement differential privacy into the training process of machine learning directly. A model trained with privacy-preserving techniques guarantees differential privacy during inference time with the post-processing paradigm (see section 2.2.1). Initial discussion started in this field of study for empirical risk minimization used in regression and support vector machine (SVM) [13], [21]. Chaudhur et al. discuss the application of noise directly in the empirical risk minimization process. This links to regularization efforts in machine learning to penalize overfitting and bounding the impact of a single sample to the training process. Abadi et al. extended this approach for deep neural networks [1] with the "moment's accountant". Instead of applying noise directly in the forward step of the machine learning algorithm, they apply noise to the backward pass that adapts the parameters as proposed by Song et al. [80]. The noise is applied either to single training samples or the mini-batches used in state-of-the-art SGD algorithms, where the privacy accountant collects the privacy-cost directly in the training steps. Similiar to Abadi et al. much work focuses on measuring the privacy exposure in SGD learning techniques more precisely to provide tighter bounds and reduce the computational overhead [5] [60].

Another group of researchers investigates more and more popular principal component analysis (PCA) techniques for privacy reasons [14], [93], [1]. PCA aims to reduce the dimensionality and complexity of data without losing the actual relevant information. This procedure can restrict the possibility to re-extract private data from data as well.

3.2 Private aggregation of teacher ensembles

One of the reasons why the PATE framework reaches a high utility during inference while guaranteeing privacy is that it does not require any perturbation to be added to the inference queries. The post-processing guarantees of differential privacy allow considering privacy aspects only during training time. Bindschaedler et al. achieve something similar by creating synthetic data sets [6] out of the actual private data before training. They describe how to mimic a data set with good utility in various statistical analyses and machine learning settings while preserving privacy. A well-equipped adversary with at least black-box access can distinguish between the real and fake data sets only up to an accuracy of 63%.
In line with the initial PATE-G approach (see section 2.3, additional research goes into generative models [68]. A GAN creates plausible samples while understanding the underlying distribution of the data set. The significant advantage of releasing a generative model of the data set over releasing just statistical information is, that there are no limitations on how the data can be processed and used. But even the representation of private data in an artificial data set requires a mechanism to quantify the privacy costs. Gergely Acs et al. provides an approach to train generative neural networks utilizing private SGD [3]. To boost the performance of their generative models, the first cluster the input data and then train specialized generative models. The "moments' accountant" from Abadi et al. [1] is used to show the models' privacy and the finally aggregated generator.
In federated learning approaches (see section 2.2.1), the aim is to keep the data locally on the user devices or the data owning party. A good overview from Li et al. can be found in [51]. Those approaches can be generalized to the idea of secure multi-party computation introduced by Yao in 1982 [90]. McMahan transferred these ideas to machine learning to learn user behavior directly on their mobile devices and only share the updated model parameters [58]. They included a first idea that such models preserve privacy by releasing only the minimum required information to a centralized party. Providing privacy in a federated setting requires that the aggregation approach guarantees privacy. Shokri and Shmatikov proposed a federated SGD algorithm for non-convex models that privately aggregate the parameters across separately trained sub-models [77]. The privacy bounds are measured by them per transferred parameter, which seems not feasible with current techniques with thousands of parameters. Besides the more technical views of the topic, there is research in secure and private protocols in a federated setting [70], [74]. Those protocols aim to provide privacy in the communication between the computation parties while preserving the utility.
Following the approach of *PATE*, Hamm et al. proposed and discussed a federated learning method based on semi-supervised knowledge transfer learning [35]. They train the final classifier on public data labeled with a private aggregated classifier. Triastcyn outlines the idea of federated generative privacy [85]. Here the distributed models act as discriminators for a generator

in a GAN setting. With the privately created artificial data, a new model can is trained. Further implementation of the PATE-G approach can be found by Jordon et al. [41]. Hayes et al. and Nays et al. discuss potential attacks against federated and GAN based privacy tools [36], [62].
As this thesis focuses on extending *PATE* for a large-scale multi-label classification one paper must be mentioned. The team around Zhu proposed an alternative to *PATE* that allows multi-label classification [94]. The approach how they aim to limit the privacy exposure by clipping the impact each teacher is debatable. The similar amount of private information are exposed just scaled down by the clipping.

3.3 Audio classification tasks

The classification of audio data is a sub-discipline of the overall classification of data. While the classification itself is not special compared to other classification tasks, audio data open some challenges to process them with classification techniques. Audio data is normally available as continuous information and must be segmented for processing [52]. Special categories of audio data as natural language use advanced techniques to address the specifics of the type of data.
A lot of effort was invested in feature extraction in the time and frequency domain from audio data [57], [52], [88], [92]. Those features were used to train simple classification models as decision trees. With the upcoming usage of support vector machines rather the whole audio segment got passed into machine learning models for classification instead of manually extracting features [33], [19]. Even with these improvements audio classification remains a challenge. A big issue is the availability of good training data. Compared to the gathering of images, building a large collection of audio samples is far more expensive.
Realistic audio event classification for urban sounds require multi-label classification which allows to assign more than one label to a sample [87]. This corresponds to the realistic scenario that multiple sound events appear in the same sample of audio data [11], [71], [12]. Various other audio classification tasks also require multi-label classification as animal detection [9], emotion detection for music [86] or music genre detection [65].
The *Audio Set* is introduced in [29]. It is a large-scale audio snippet collection with multi-label tagging of the audio events. There are various works around this data set. The Google Brain team set the initial baseline in the original paper with a mAP value of 0.314 [29]. A similar team around Shawn Hershey argues that CNN are the most effective technique for audio processing on a predecessor of the *Audio Set* (see section 4.1) [39], [43].
The Detection and Classification of Acoustic Scenes and Events (DCASE) challenges are yearly public research challenges [18]. They often build on parts of the *Audio Set* data set and aim to solve specific sub-problems, e.g. the sparsely labelling and background noise, within the data set. Changsong Yu et al. made the first significant improvement for the *Audio Set* with a multi-level attention model [91]. Instead of using only the final layers of the deep neural network structure, the outputs of the intermediate layers are considered for the classifier. This pushed the mAP benchmark up to 0.360 [29]. The research group around Qiuqiang Kong published various papers with improvements for classifiers on the *Audio Set* [45], [48] and [46]. In his recent article, they introduced *PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition* [47]. These models utilize transfer learning from other audio classification

and visual classification tasks. In contrast to most other works, they have not used the published embeddings from the *Audio Set* downloads but the actual audio snippets. With the original paper Gemmeke et al. published pre-processed representations as embedding for the *Audio Set* [29]. Kong et al. fed the raw audio signal with pre-processed log-Mel diagrams to the classifier. All those improvements pushed the mAP score to 0.439, the current *Audio Set* tagging systems benchmark. Similarly, Logan Ford et al. achieved a mAP of 0.392 with a classifier on the raw audio log-Mel data without transfer learning [27].
Similar data sets to the *Audio Set* are the ESC-50 [72] and the Urbansound8k [75] sets. The first one contains 2000 samples distributed among 50 classes. In comparison, the Urbansound8k has about 8000 examples out of 10 classes. These data sets are not comparable to the *Audio Set*, which is significantly larger.

4 Designing and building an audio classifier under consideration of PATE

Building a classifier for audio data is still a challenge in machine learning. The aim is to learn how to classify sounds and predict labels for a sound event. There exist various practical applications as assigning a genre to a music clip, detecting anomalies in continuous noise, or the detection of words in natural language processing.
This work aims not to build any audio-classifier but to create a privacy-preserving audio classifier under consideration of the *PATE* framework. There exist no working end-to-end implementation of the *PATE* framework that fits this use case of audio data and multi-label classification. Hence the transfer of the *PATE* framework to support these properties is custom.
The actual model training is similar to other supervised training approaches. Except that not one big model utilizes all the available data for training, and instead, several smaller teacher models are trained. To split the information, accordingly custom data sets are randomly created. The actual goal of teaching a private student model is achieved in a aggregation step. It utilizes the predictions of the teacher models while the privacy consumption is measured for every query to the ensemble.
Classification models are trained based on large amounts of labeled audio snippets. They can then label audio clips of the same distribution during inference. One of the most promising architectures that can understand the complexity of audio data are CNN architectures [39]. In this work, a lot of standard machine learning tools of the Pytorch framework are used. For the build and training loop of such a classifier, the required steps are described in this chapter.
The labeled raw audio files need to be pre-processed for the machine learning model. The pre-processing step creates an image-like representation of the sound snippet or even combines multiple extracted features of the audio data. Common hurdles as noisy background, ambiguity, and complexity of sound events, and how humans recognize audio make audio classification a complex field of research. The art behind audio data is to extract representative and meaningful features from the raw data. There are many ways to manually describe audio data in the time and frequency domain. The extracted properties can be used for the machine learning tasks. Alternatively the raw audio data can be utilized for the training. This set of techniques plays a minor role in this thesis. Instead the main work is focused on the privacy-preserving training approach.
This chapter starts to outline the relevant characteristics of the used data collection *Audio Set* in section 4.1. In section 4.2 the original *PATE* aggregation mechanisms and privacy analysis is extended to support multi-label classification. The practical build of a classifier for city and environmental sound under consideration of the *PATE* framework is further explained in section 4.3. The remaining section 4.4 outlines features to overcome some problematic characteristics of the *Audio Set* discussed in section 4.1.

4.1 Audio Set data set

The *Audio Set* data set is a huge publicly available data set of audio event snippets [29]. The overarching goal of the authors was to narrow the gap between image and audio classification by providing a big experiment data set. For image tasks a wide range of data collections exist, but audio data is much harder to prepare and hence more expensive. They designed first an ontology of 632 hierarchical classes for general audio classification. All samples in the data set are ten seconds long and is represented by a YouTube video id with a start and end timestamp. Each audio clip from the YouTube video can have one or multiple labels from the ontology assigned. Human raters labeled the videos with audio and visual data present. The setup presented no metadata or title to the raters. Votes were collected by three people per sample. Only if the majority of the raters agreed the video id got the label assigned. As reported by the authors, $76,2\%$ of the votes were unanimous and 23.6% with a 2:1 majority [29]. The degree of consensus between human raters differed per class.
To gain a high number of samples per class, some pre-filtering was required which proposed only clips that are likely to be labeled with the respective class. They used a Google internal video classifier that works on metadata, engagement signals, and the video content itself for this task. In addition, plain text, pattern searches, and nearest neighbor approaches were used to find enough samples per class. The released data set contains two million samples (more than 5.000 hours) for 485 audio event classes. The authors excluded some classes as they were too complex to classify by a human or ambiguous in the understanding.
The authors themself set up a baseline for classification tasks on the *Audio Set* data. Instead of using the ten seconds samples directly, they used one second chunks and aggregated the results of the classifications. The baseline has a mAP of 0.314 and an average AUC of 0.959. They report already that the performance differs among the classes. More recent baselines come from Kong et al. with a mAP of 0.439 [47] and Gong et al. with a mAP of 0.474 [32].
For the work of this thesis, a subset of classes from the *Audio Set* ontology is used. The classes were limited to audio events related to city and environmental sound. The thesis used about one million of training samples across 150 classes.

4.1.1 Ontology of the Audio Set

Together with the data set, the authors from Google invested effort in designing a sustainable ontology for audio events. Their intention during the design was to build up a structure that makes it simple for a human to perform audio classification. A hierarchical and non-exclusive set of classes allows a rater to find a label representing the heard audio snippet. If a class is too specific or the audio sample is ambiguous, the rater can navigate up in the hierarchy and assign a more generic label. Limiting the depth and the number of child nodes lets a person quickly navigate through the tree.
They conducted a large-scale analysis of web texts for an initial sampling of meaningful labels. This reduces the impact of an individual opinion on a good or bad descriptive word for a sound event. Their algorithm performed analysis to find hyponyms of the term “sound”, which then are used as sound classes. All classes must fulfill defined properties.
A category must correspond to the idea or understanding that immediately comes to a person’s

mind listening to the audio sample. To keep the data set general, no splits of classes are allowed that can be only distinguished with expert knowledge in the respective domain. This could be the separation between specific animal subspecies or more opinion-based categories as music subgenres. Visual descriptions are avoided in the class names. For example, 'Walking on leaves' refers to an audiovisual understanding of the sample, that is not intended to be in an audio event ontology. Instead, only audible representations are used.
For this master thesis, the available classes were manually restricted to sound events in a city and environmental context. Sound that can appear externally while driving or riding through a city is used as selection criteria. The root node "music" of the ontology and all subnodes are entirely removed as music classification is a different use case. Instead, the focus of the manually selected classes is on "sound of things". In total, this work uses about 150 classes. Another smaller subset of 15 classes is used for demonstration purposes. For this smaller experiment set, easy and hard to classify labels were picked after the first classifiers trained. Easy classes have a detailed sound profile as 'church bell' or "jackhammer" while more challenging classes can have a lot of ambiguous sounds like 'bus' or 'train'. The final characteristics of the experiments are outlined in section 4.3.1.

4.1.2 Characteristics and limitations of the Audio Set data

To work correctly with the *Audio Set*, it is essential to understand some data characteristics. These properties impact the distribution of labels across the data and affect the quality of the labels for different classes. The techniques used are described in section 4.4.
It is worth noting that the data set is highly imbalanced regarding the different classes. Some classes as "Speech" and "Music" have more than a million annotations, while others only have a few hundred samples. This inherits from the domain the data was collected and the structure of the ontology. Youtube videos mostly contain either speech or music, and the generic classes are easy to map for a human person labeling the samples. This property requires techniques that address this imbalance during the training of the classifier.
An annotation within the hierarchical ontology does not require that all parent classes are mapped in addition if a subclass is mapped. This opens space for interpretation for the person annotating the sample. There might be cases where the sample is mapped to a path in the tree without high confidence that it is correct, or the person was too lazy to pick all parent classes. There are no explanations within the *Audio Set* paper for this circumstance.
According to the ontology design, a reader can assume that all parent classes are applicable if a child class is assigned to a sample. This can sometimes be conflicting as a small set of labels appears multiple times on different positions in the ontology.
Another point of relevance is the quality of the assigned labels. This mainly results from how the labels are given to the video samples. The in addition to the audio data presented visual elements of the video is one considerable aspect for the distortion of the assigned labels. This second channel of information distracts the person during the labeling process. For example, even if there are no audible sounds of a car but its visual representation in the video, the person might assign the car label anyway.
Additionally, it is complex to pick all relevant classes out of an ontology of more than 600 classes. The team behind the *Audio Set* conducted a quality assessment and a re-rating of up

to 1000 samples per class to quantify this issue. Out of this data review, they calculated quality estimates per class. About half of the classes have a quality estimate with less than 70% [82]. This means that only seven out of ten samples labeled with the class are correct representations of the class. The size of the data collection should compensate for this poor quality.

4.2 Adaptation of PATE for multi-label classification

The original *PATE* framework is built for multi-class classification. As aggregation of the teacher ensemble prediction, a histogram outputs the class with the highest number of votes among the teachers. Similarly, the privacy analysis builds upon the idea that a single class wins upon all others. While the general intuition of the *PATE* approach can be applied to multi-label scenarios as the *Audio Set* classification problem, some adoptions are required.
In a multi-label setting the aggregation result is presented as a vector with a vote per class. The averaged prediction vector $m \in [0,1]^n$ of all teacher predictions shows the result. A value close to zero corresponds to all teachers assigned a low probability for a class and a one the other way around.
The amount of teachers in the ensemble does not matter. This assumes each teacher equally contributes to the aggregated vote per class in $[0,1]$. Hence, the threshold describes how significant the average probability for a class over all teachers must be to consider the sample a class member. Every teacher can flip the boolean prediction for each class if there is no strong consensus among the teachers. This applies to small ensembles up to a single teacher and as well for large ensembles. Even if the individual teacher's impact decreases with the size of the ensemble, every teacher equally contributes to the aggregated result.

4.2.1 Threshold aggregation mechanism

The new aggregation can not directly transfer the basic histogram approach to a multi-label scenario. Instead of a single class that wins over all others, an undefined count of classes can now be considered present by the ensemble. A multi-label aggregation mechanism takes the multi-label classification of the teachers as inputs and outputs a set of ensemble labels for this sample. There is no information about the not-assigned labels publicized. The raw list of assigned labels is used later for the student training to keep the privacy guarantees. Below two aggregation mechanisms are defined.

Definition 10 (GNThreshold)**.** *For the Gaussian noise threshold (GNThreshold) aggregation, all teachers' probability predictions per class are average. The average is passed into a noisy threshold check that transforms the aggregated probabilities into boolean statements.*

$$\mathcal{M}_{\sigma,th}(x) \triangleq c \in \{0,1\}^n, \text{ where } c_i \begin{cases} c_i = 1, m_i(x) + \mathcal{N}(o,\sigma^2) > th \\ c_i = 0, m_i(x) + \mathcal{N}(o,\sigma^2) \leq th \end{cases} \tag{4.1}$$

The GNThreshold utilizes the average m of all teacher predictions per class i. It applies a threshold to the noisy average for each class i to map them to a positive or negative predictions for the class. The algorithm outputs a binary vector c with a size corresponding to the number of

classes. The value of c_i is 0 if the class is not detected and 1 otherwise. All 1 predictions lead to a sample label with the class. In alignment with the original *PATE* framework Gaussian noise is added with mean 0 and variance σ^2. It is unnecessary to scale the noise according to the number of teachers. The averaging of the teacher predictions already scales the votes accordingly. A variance of σ^2 denotes that the applied noise disturbs the result regularly by σ^2 teacher votes.
An issue with threshold algorithms is that they open up a new parameter that must be estimated or trained. The threshold corresponds to the teachers' ratio that agrees that the ensemble should assign this label. Partial votes in case the teachers are unsure can be considered. Those teacher votes in general, and especially partial votes, are not published.
The performance, as well as the privacy budget, depends on the threshold. If the required agreement of the ensemble is selected too low or too high, either label assignments are missed or become incorrectly assigned. Privacy of the *PATE* framework intuitively depends on the probability that the most likely outcome will not be outputted. Adapting the threshold impacts the distance that allows to flip a vote. The reader can find further discussions on this in section 4.2.3.
To improve the threshold further, the algorithm can select it adaptively for each sample. Instead of a fixed threshold for all samples, the threshold is chosen based on the teacher votes. The assumption is that positive predictions have a strong consensus among the ensemble while negative predictions have a low score. In a perfect scenario, all teachers vote with maximum probability for the positive classes and zero probability for the negative classes. If the threshold is exactly between those two extremes, a single teacher's probability flips the vote becomes minimal.
For example, there is an ensemble with ten teachers and five possible classes. The ensemble predict a sample and the aggregated votes look as follows votes $= [0.0, 0.1, 0.8, 1.0, 0.0]$. For a fixed threshold of $th = 0.7$, teachers must agree in their vote to consider a class as present in the sample. The outcome for the class prediction with eight votes highly depends on a single teacher. If only one vote flip, the outcome can change for this class (the aggregated votes must be greater than the threshold).
With the adaptive technique, it is possible to choose the threshold according to the distribution of the votes. The biggest gap between the ordered list of votes is between the classes assigned with 1 and 8 votes. Selecting a threshold in the middle of those two class vote counts reduces the single teacher's impact on the result. With an adaptive threshold of $th_{\text{adaptive}} = 0.45$, it requires 3.5 teachers to flip their votes until it has an impact on the result.

Definition 11 (GNThreshold Adaptive)**.** *For the GNThreshold Adaptive aggregation, the probability predictions per teacher are averaged for every class. Let $k_{i,j}$ be the descending ordered list of aggregated probability predictions with i as the class id (position in the original vector) and j the index of the ordered list. The maximum adjacent difference $d = max_j(k_{a,j} - k_{b,j+1})$ in the ordered list of aggregated probabilities is determined. The average is passed into a noisy threshold check that transforms the aggregated probabilities into the boolean result vector.*

$$\mathcal{M}_{\sigma,th}(x) \triangleq c \in \{0,1\}^n, \text{ where } c_i \begin{cases} c_i = 1, m_i(x) + \mathcal{N}(o,\sigma^2) > k_b + d/2 \\ c_i = 0, m_i(x) + \mathcal{N}(o,\sigma^2) \leq k_b + d/2 \end{cases} \tag{4.2}$$

The Gaussian noise adaptive threshold (GNThreshold Adaptive) aggregation method formalizes this intuition. The distance to the threshold value should be maximized as an overarching goal to reduce the probability of a flip in the final votes. If two similar maximum adjacent differences exist in the ordered set, the higher aggregated probabilities are chosen.
This approach becomes complicated if the teacher ensemble can not agree strongly one some classes. In such a case the fixed threshold might address the problem better.

4.2.2 Sensitivity of multi-label data

Both definitions for differential privacy and RDP build upon the idea to bound the privacy impact of a single sample in the data. The intuition of the privacy techniques is to mitigate the effects on the final function of an individual sample. Sensitivity quantifies the impact of this sample on the outcome of the procedure. This is formulated in the definition of global sensitivity in section 8. The sensitivity is a driving factor for the privacy cost of the algorithm, as discussed later in section 4.2.3. To transfer the *PATE* privacy analysis to a multi-label scenario, the question of the sensitivity of a query is elementary.
The information contained in a binary multi-label sample of shape $\{0,1\}^n$ corresponds to the number of classes available, with n being the number of classes. Compared to the multi-class classification, the information contained in a sample grows by $n-1$. As a single sample in the worst-case scenario flips all binary class predictions, the global sensitivity is $GS(f, \|\cdot\|_1) = n$. Multi-label classification is similar to performing n-times binary classification. Each of them specify if a class is assigned or not. This analysis can be improved by using the L2-norm to model the worst-case outcome. In any case, this leads to extremely high sensitivity values causing insufficient privacy guarantees.
In the here considered multi-label classification, the label vector is expected to be very sparsely filled. The average amount of labels assigned within the *Audio Set* is about 2. Intuitively it follows that the sensitivity should be far less than the number of classes. Various possible approaches are promising.
The initial idea is to switch to local sensitivity instead of global sensitivity [34]. Local sensitivity, without defining it here formally, describes the maximum difference for all neighboring data sets from one particular set of data. It is often used as global sensitivity explodes. The global approach instead models the maximum difference according to all possible data sets. This could bring a considerable advantage as the characteristics of the sparsely labeled *Audio Set* could be considered in calculating the local sensitivity. Unfortunately, local sensitivity requires a different approach in regards to privacy analysis. There is currently no approach available to analyze RDP with local sensitivity.
Another approach is the concept of τ-approximation is used to clip the impact of an individual sample [94]. The idea here is to clip each sample as it can only vote for maximum τ classes. τ specifies the expected amount of assigned labels for all samples. To clip the prediction with τ-approximation the binary vector $f(x) \in \{0,1\}^c$ gets scaled as follows.

$$\hat{f}(x) = f(x) \cdot \min\left(\frac{\tau}{|f(x)|}, 1\right) \tag{4.3}$$

This clipping allows to bound the global sensitivity tightly. In the extreme scenario where all class predictions flip the sensitivity with respect to the L1-norm of $GS(\hat{f}(x), \|D - D'\|_1) = \tau$. This approximation would allow to use a lower sensitivity during privacy analysis. It could be argued that this holds formally for the sensitivity, but practically it reveals the same amount of information as long as binary multi-label classification is used. Every class that gets a score greater than zero indicates a positive classification. Hence this approach is rather not suitable for this task.
The approach for the *Audio Set* is much simpler. While creating the label assignments, three options are presented 'present', 'unsure', and 'not present'. For the final release, only the 'present' information got published. Hence there is no guarantee that a class is not in the sample. It is not the goal to make a decision for all classes, but to assign the most matching labels to a data point.
The sensitive information that must be protected is just the positive label assignments and predictions for all possible classes. With this information, the sensitivity of a query can be bounded by the impact a single sample can have to flip from false to positive for a class prediction. The *Audio Set* has on average 1.98 labels assigned per sample. The aim should be to have a sensitivity close to the average count of labels. As for privacy considerations, the average is insufficient in a worst-case scenario that must be assumed.
Instead, the sample with the highest count must be considered for the worst-case scenario. To not penalize the whole privacy analysis due to one outlier with an increased number of label assignments, such samples are dropped. For further privacy analysis, it can be assumed that the impact of a single sample on the output of the classifier is bounded to the maximum assigned labels for any sample used during training.

4.2.3 Multi-label PATE privacy analysis

The idea of the privacy analysis can be applied from the original paper. Significant changes must be made to respect the changed global sensitivity through all steps of the privacy analysis. For this, the estimation of the probability that the most likely outcome will not be outputted must be reassessed. In the below, the data-independent analysis is migrated to a multi-label scenario, and afterward, the changed approach for the data-dependent analysis is discussed. Multi-class statements mainly refer to the appendix of the *PATE* papers and are not described in full detail here. For proof, please refer to the papers. Instead, the multi-label privacy analysis is its work and is described extensively.
For the data independent analysis, the information about the actual sample queried can not be used. Hence only generic properties of the aggregation can be used. The multi-class mechanism of the original *PATE* paper fulfills $(\lambda, \frac{2\lambda}{2\sigma^2})$-RDP (see section 2.3.3). This builds upon the RDP privacy guarantee of the Gaussian mechanism from Mironov [59] and the histogram of the aggregated votes.
To flip the outcome corresponding to the maximum vote count, the classes with the most and second-most must be considered. The necessary change must be bound for the privacy analysis that the second most outperform the current winning class. As the Gaussian mechanism is applied to all classes individually, this estimation has the doubled variance. RDP allows to add up privacy values for the same order, which is discussed in section 2.3.3.

To transfer this into a multi-label scenario, no bigger adaptions are required. For multi-label, not only one class is relevant, but all classes. A multi-label classifier result can be seen as an n-time prediction of boolean values that indicate if a class is present or not. This applies to the privacy analysis as well. That each teacher can flip the final outcome in any class must be considered for privacy. The data-independent privacy for the multi-label scenario can be estimated as below:

Theorem 12. *The GNThreshold mechanism fulfills the data independent* $(\lambda, \frac{n\lambda}{2\sigma^2})$*-RDP for all* $\lambda \geq 1$*.*

The amount of the Gaussian mechanism corresponds to the count of classes. In a worst-case scenario, a single teacher flips all boolean predictions. The aggregation induces Gaussian noise to the averaged values before applying the threshold. This algorithm can be interpreted as multiple Gaussian mechanisms that fulfill RDP guarantees. With this the $(\lambda, \frac{n\lambda}{2\sigma^2})$-RDP is claimed. The exposed privacy depends on the number of classes n and the applied noise σ^2. The noise factor can be interpreted as how many teacher votes are randomly added to the ensemble prediction. Hence the value of the noise *sigma* should be searched around half of the number of teachers in the ensemble. Otherwise, the random noise dominates the result.
This approach for data-independent privacy analysis results in extremely high dependence on a single teacher and sample. The RDP privacy cost scales linearly with the number of classes. With an increased number of classes, only a few queries to the ensemble are possible until the whole privacy budget is spent.
Observing the data allows tightening the privacy analysis. Not all class predictions are similarly likely to flip their vote. If all teachers agree that a class must be assigned to a sample, the impact of the presence of a training sample on the overall result is low. This training sample can impact only one teacher. If the number of teachers is high enough, the decision of a single teacher becomes irrelevant.
The intuition of a probability that describes how likely the overall result is to change is used already in the original *PATE* data-dependent privacy analysis. There is a likely outcome event i^* of the aggregation algorithm. The probability $Pr[\mathcal{M}_\sigma(D) \neq i^*] \leq \widetilde{q}$ describes how likely it is that this outcome will not be outputted for the sample. Based on this estimation, a tighter privacy analysis can be provided.
To transfer this idea to a multi-label scenario, some adaptions are required. It starts with estimating the probability of the likely outcome q. Compared to a multi-class scenario, the discrete event space of possible outcomes increases. A boolean value per class exists instead of just the number of classes n now, which blows up the event space to 2^n.
Further, a likely outcome $i^* \in \mathbb{R}^m$ for a multi-label scenario with the great support of the teacher ensemble must be evaluated for a threshold. This deviates from the original idea of the histogram in a multi-class scenario. The outcome does not depend on the other class aggregated votes but directly on the outcome for each class. It is no longer evident by the majority which class wins or not. Instead, the distance to the threshold decides if there are enough votes assigned to a class or not. Obviously, the higher the threshold is, the stronger the consensus of the ensemble for each class must be.

Theorem 13. *For a GNThreshold aggregator $\mathcal{M}_{\sigma,th}$ in a multi-label scenario, the aggregated teacher votes $\bar{s} = (s_1, \ldots, s_m)$ and for any $i^* \in \{0,1\}^m$ there is*

$$Pr[\mathcal{M}_{\sigma,th}(D) \neq i^*] \leq q(\bar{s}) \tag{4.4}$$

where

$$\begin{aligned} q(\bar{s}) &= 1 - Pr[\mathcal{M}_{\sigma,th}(D) = i^*] \\ &= 1 - \prod_{i \in m} F(|s_i - th|; 0, \sigma) \end{aligned} \tag{4.5}$$

with F representing the cumulative distribution function with variance σ^2 and mean 0.
The $Pr[\mathcal{M}_{\sigma,th}(D) = i^*]$ represents a specific outcome and the corresponding probability that the randomness of the mechanism flip any of the boolean outputs for a class.

To compute the probability that the overall prediction does not come true as described in theorem 13, each class must be reviewed separately. The chance that the output flips for a single class prediction depends on the difference between the aggregated teacher votes and the threshold in correlation to the noise applied.
This can be calculated similarly to the multi-class scenario using the error function or the CDF. Instead of the delta to other class scores, the total difference between the threshold and the noise is used. This addresses the fact that only an extremely positive or negative amount of noise lets the overall prediction for the class change. If the applied noise exceeds the difference, it must be considered a flip. As the gap increase, it becomes more unlikely that the applied randomness will change the outcome.
It helps the analysis that only positive or negative noise must be considered. If the aggregated votes are below the threshold, negative noise value can not impact the outcome and vice versa. Each class outcome can flip individually. Therefore those events are considered as independent and multiplied. As a perfect result, all teachers agree on the votes, which minimizes the privacy budget spent. The GNThreshold Adaptive mechanism maximizes the total difference between the vote values and the adaptive threshold. This allows a smaller $\tilde{q}$ while just considering the data itself.
The different classes are considered independent for further analysis, even if that is technically and practically not the case. As the correlation between the outputs derives inherently from the machine learning models, it must be correlated. Otherwise, it would be a multi-class scenario.
Under this assumption, the probabilities for each class must be multiplied as described in theorem 13. This overall probability describes how likely it is that a prediction of the ensemble will not be outputted. It can range from 0 to 1, where 0 corresponds to overwhelming confidence of the ensemble and one the opposite. For large values of $|s_i - \text{th}|$ the cumulative distribution function gives numbers close to 1. Multiple values close to 1 result in a product close to 1 as well. As this value is subtracted from one, it turns into a small estimate for $\tilde{q}$ that corresponds to a confident prediction.
For the actual calculation of RDP guarantees, most explanations and proof of the paper are applicable. The idea can be found in 2.3.4 and in theorem 6 of the original paper [67]. The idea is to find a function that describes for the randomized algorithm $\mathcal{M}$ the RDP based upon other

higher moment RDP guarantees and the value of $\tilde{q}$. For this approach, two higher-order RDP guarantees with order μ_1, μ_2 are used. They become translated into a new RDP guarantee of order λ whereas $\lambda \leq \mu_1, \mu_2$.
The proof bases on the discreteness of the output space and the possibility to disassemble and estimate the values for the real q. The output is still discrete. For the proof, the sequence of boolean outputs must be understood as an event that is considered. This makes it simple to describe and estimate all possible output sequences.
The only adaption required is calculating optimal higher moments μ_1 and μ_2. Those are used as parameters for the final equation later in this section. As the increased global sensitivity must be considered here, minor adaptions to the original paper are required.

Theorem 14. *To minimize the term* $\boldsymbol{A}$ *of the equation on equation 11 for variance* σ^2 *below optimal higher-order for* μ_2 *is used. It scales according to the sensitivity of the aggregation mechanism.*

$$\begin{aligned}
\log\left\{(\tilde{q}e^{\epsilon})^{1-\frac{1}{\mu_2}}\right\} &= \log\left\{\tilde{q}^{1-\frac{1}{\mu_2}}\exp\left(\epsilon_2\cdot(1-\frac{1}{\mu_2})\right)\right\} \\
&= \log\left\{\tilde{q}^{1-\frac{1}{\mu_2}}\exp\left(\frac{n\mu_2}{2\sigma^2}\cdot(1-\frac{1}{\mu_2})\right)\right\} \\
&= (1-\frac{1}{\mu_2})\cdot\tilde{q}+\frac{n\mu_2-n}{2\sigma^2} \\
&= \frac{1}{\mu_2}\log\frac{1}{\tilde{q}}+\frac{n\mu_2}{2\sigma^2}-\log\frac{1}{\tilde{q}}-\frac{n}{2\sigma^2}
\end{aligned} \tag{4.6}$$

which is minimized at $\mu_2 = \sqrt{\frac{2\sigma^2\log(1/q)}{n}}$.
To minimize the second term $\boldsymbol{B}$ *the term* $e^{\epsilon_1}/\tilde{q}^{1/(\mu_1-1)}$ *is minimized.*

$$\begin{aligned}
\log\left\{\frac{e^{\epsilon_1}}{\tilde{q}^{1/\mu_1-1}}\right\} &= \log\left\{\tilde{q}^{\frac{-1}{\mu_1-1}}\exp\left(\frac{n\mu_1}{2\sigma^2}\right)\right\} \\
&= \frac{n\mu_1}{2\sigma^2}+\frac{1}{\mu_1-1}\log\frac{1}{\tilde{q}}
\end{aligned} \tag{4.7}$$

which is minimized at $\mu_1 = 1+\frac{\sqrt{2}\sigma\sqrt{\log(1/\tilde{q})}}{\sqrt{n}} = 1+\mu_2$.

Putting all the above explanations and the work from the original paper together results in the following steps to perform an RDP analysis. As a result, a RDP value of a specific order λ is outputted. To start, the value for $\tilde{q}$ is calculated according to the theorem 13. Now the data-independent and data-dependent privacy analysis are calculated. For simplicity, the minimum of those two bounds is outputted. It is possible to calculate an extreme value q_0 that allows separating if the data-dependent or data-independent calculation is always beneficial.

$$D_\lambda(\mathcal{M}(D)\|\mathcal{M}(D')) = \min\left\{\frac{1}{\lambda-1}\log\left((1-\tilde{q})\cdot\boldsymbol{A}(\tilde{q},\mu_2,\epsilon_2)^{\lambda-1}+\tilde{q}\cdot\boldsymbol{B}(\tilde{q},\mu_2,\epsilon_2)^{\lambda-1}\right), \frac{\lambda}{\sigma^2}\right\} \tag{4.8}$$

For the data-dependent calculation the formula out of 11 is used. The values for μ_1 and μ_2 are chosen according to theorem 14. For the higher order bound required for the data-independent the theorem 12 hands $\epsilon_1 = n\mu_1/2\sigma^2$ and $\epsilon_2 = n\mu_2/2\sigma^2$ which still depends on the global sensitivity of the mechanism.
Some conditions from the original paper can be taken over to decide if it is worth to evaluate

the first equation of the formula. The first part of the formula must be evaluated only if $q \leq 1$, $\mu_1 \geq \lambda$, $\mu_2 > 1$ and $\tilde{q} \leq e^{(\mu_1-1)\epsilon_2} / \left(\frac{\mu_2}{\mu_1-1} \cdot \frac{\mu_2}{\mu_2-1}\right)^{\mu_2}$. All those expressions lead to extreme values of the formula that are not desired. There are no further adaptions required for a multi-label scenario as the increased sensitivity is already considered in the ϵ and μ calculation.
The optimal λ is not known upfront. Only in the moment of translation into (ϵ, δ)-DP an optimal λ must be chosen (see theorem 7). Hence for the practical implementation, a wide range of possible values of λ are computed in parallel by calculating over a logarithmic array of orders λ. To enable deeper analysis, all privacy-related values are logged in the implementation. It is sufficient to store only the final values privacy cost for a practical application often already transferred to (ϵ, δ)-DP.
The privacy analysis for a multi-label sample is a function of the results of the aggregation mechanism, the used threshold, and the applied noise. All those elements influence the exposed privacy. In general, the exposed privacy is less if there is a strong agreement between the teachers. This can be seen as high distances from the threshold. Unfortunately, guaranteeing strong privacy is more complicated than in a multi-class scenario. Instead of a high peak in a histogram, it requires a strong consensus among all classes to show strong confidence.

4.2.4 Confidence aggregation mechanism

The confidence GNMax algorithm of the original paper is described in 2.3.6. The aim of adding a confidence barrier to the aggregation is to reduce privacy consumption if the ensemble has a weak consensus. If the teachers strongly agree, the privacy consumption should be less and vice versa. While analyzing if the teachers agree strongly, additional privacy budget are consumed. The trick of the confidence aggregator is to compare the results to reduce actual privacy consumption while keeping the extra privacy cost for the confidence check as little as possible. This is mainly achieved by applying more noise to the confidence check, but this is not the only adjustable setting in a multi-label scenario.
For the confidence check, the raw aggregated teacher votes m_{raw}, another confidence threshold th_c, and a noise parameter σ_c are required.

Definition 12 (Confidence GNThreshold)**.** *For the confidence GNThreshold aggregation, the confidence of the top n classes is checked against a threshold th_c. If all those elements pass the check, the regular GNThreshold result returns. Alternatively, no answer is returned for this sample. The parameter n describes how many elements of the raw aggregation should be checked.*

Algorithm 3 Confident-GNThreshold Aggregator

Input input m, confidence threshold th_c, teacher ensemble threshold th_e, noise parameter σ, a number n, and confidence noise parameter σ_c

1: **if** $\forall j \in n$ max elements δ_j of $\delta = m_{raw,j}(x) - th_e : \delta_j + \mathcal{N}(0, \sigma_c^2) \geq th_c$ **then**
2: **return** $GNThreshold(m, th_e, \sigma)$
3: **else**
4: **return** $\perp$

For the confidence check, the noisy distance to the threshold for each class are compared to a pre-defined confidence threshold. The applied noise is usually chosen as $\sigma_c \gg \sigma_e$ to address that

this check is performed for every sample. The actual GNThreshold is only performed if the result passes the threshold check. Even if that is not the case, the privacy spent for the confidence check is consumed anyway. The other way around, if the regular GNThreshold prediction is performed, there is no additional privacy consumed by the confidence check.
Revealing the same data twice does not harm privacy again. Instead, only the consumed privacy of the regular GNThreshold is added up. The privacy cost mainly depends on the label information used during the calculation. Hence limiting the number of labels, the calculation is performed with reduces the consumed privacy. This is why introducing the parameter n represents the top n aggregated votes. The top n operation itself is a function on the data that does not consume privacy as it does not reveal any information about the data. A list of numeric values can be ordered and then split after the n-th element without revealing any information about the values themselves. Both aspects allow having a very tight privacy consumption for the confidence check. The confidence mechanism can outperform the regular GNThreshold aggregation privacy-wise if the additional privacy cost for the confidence check is less than the saving in privacy cost for the actual aggregation.
This new parameter n should be neither picked too low nor too high. If it is too low, only the maximum predictions that are strongly above the threshold are considered. Instead, the ones close to the threshold with a high probability to flip are ignored. The other way around observing too many labels results in a higher privacy cost. As the expected maximum of assigned labels is three, the values worth considering should be searched slightly above this number. The top four elements are observed for the confidence check in this work.
Privacy-wise, the confidence check can be analyzed similarly to the regular GNThreshold privacy. It is possible to reflect the data independently or with considering the actual data. The huge difference here is that not all classes must be observed for privacy, but only the small portion used within the threshold check.

4.3 Training of a PATE based classifier

The overall effort of this work is to build an audio classifier in a privacy-preserving way. After outlining the privacy aspects and the adaptation of the *PATE* framework, the focus is now on the classification task itself. The pipeline of building an audio classifier for the *Audio Set* under consideration of *PATE* is shown in the figure 4.1.
The *Audio Set* requires several preparatory steps that are shaded in blue in the figure. It starts with the *Audio Set* data and the video platform Youtube. For the thesis, different subsets of the overall *Audio Set* collection are prepared. This is further discussed in 4.3.1. A crawler downloads the audio snippets for the prepared data collections. The raw audio data are the inputs to the pre-processor further described in the section for the audio set 4.3.2. The outputs of this step are prepared log-Mel spectrograms that will be used to train the models.
Audio classification and the *PATE* framework have specific requirements for the data fed into the training process. The data curators must split the available information across the teachers and student model to allow the privacy mechanism to work correctly. Instead of building a single classifier based on all available data, the *PATE* framework requires a unique construction of the

final classification model. The teachers and the student are individually trained classification models on the same domain of data. Each of them serve a specific purpose within the *PATE* framework.
The training of the teacher models is illustrated in green. All teachers of the same experiment together build the ensemble. The teachers are trained on a disjointed set of data. This step is further explained in the section 4.3.3.
For further steps, each teacher in the ensemble predicts the samples of the student train data. Only the created predictions and the student train data itself are required to train the student model and the privacy analysis. These steps illustrated in red are discussed in sections 4.3.4 and 4.3.5.
Most importantly, all intermediate steps like the teacher models can be deleted after the student model is successfully trained. During inference, only the element below the dashed line in the figure 4.1 are required.

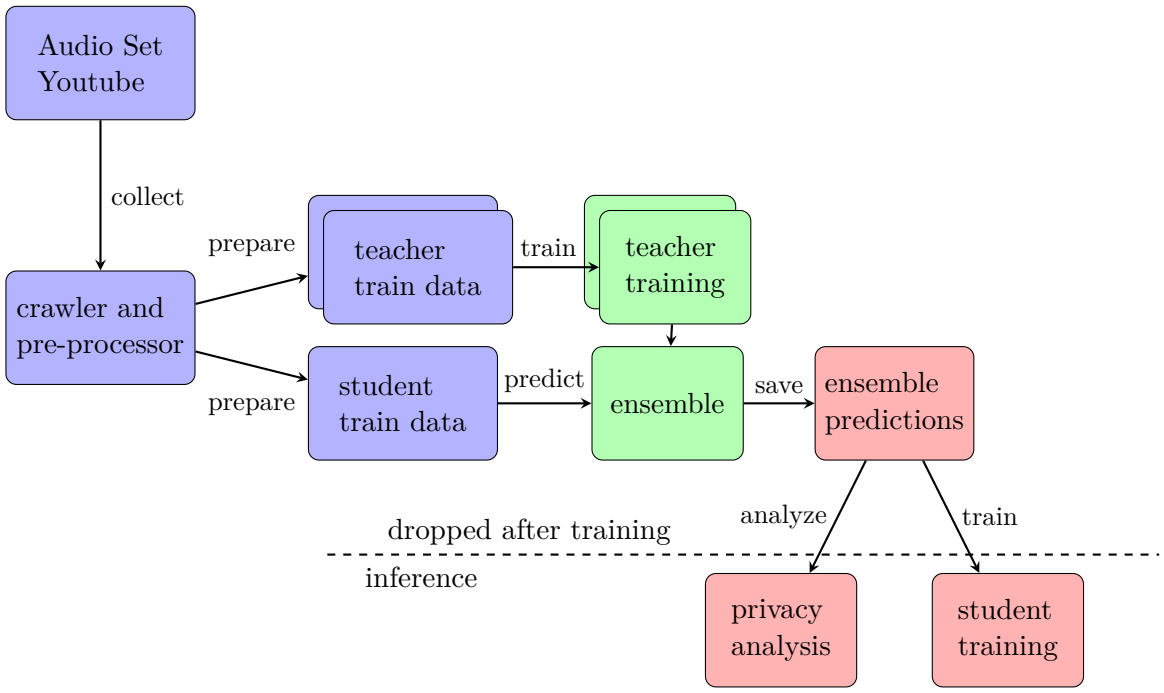


Figure 4.1: Pipeline to build an audio classifier for the *Audio Set* under consideration of *PATE*.

4.3.1 Data set preparation

The *PATE* framework assumes that the data is naturally split across different parties that do not trust each other. These parties train their teacher models individually on their data. Additionally, a separate non-private data set must be available for the student's training.
As this separation is not naturally given by the *Audio Set*, a data set generator is built that randomly creates these disjoint sets for the teachers and the student and prepares them for the training loops. This includes statistics, limiting the relevant classes, and making random subsets for the teacher and student. These data sets are the foundation for the evaluation section 5.
To validate the performance of the machine learning models, a test set is required that is not shown to the classifier during training time. An official evaluation set exists for the *Audio Set* that allows comparison against other work [29]. It contains at least 59 examples per class. Most

classes have more samples as each sample can have more than one label. All evaluations for teacher models and student models build on this evaluation set. If some classes are not available within the trained model, they are removed from the evaluation set accordingly. This test set is used for any evaluation task in this thesis which allows comparing different stages for their performance.
In table 4.1 the statistics of the generated data sets can be found.

data set	**baseline**	**small 10**	**small 20**	**small 50**
Overall size	95696			
Class number	15			
Overall teacher train s.	76656			
Individual teacher train s.	76656	7655	3827	1531
Student train size	19139			
Test size	1174			
data set	**baseline**	**big 10**	**big 20**	**big 50**
Overall size	342979			
Class number	153			
Overall teacher train s.	274383			308681
Individual teacher train s.	274383	27438	13719	6173
Student train size	68595			34297
Test size	5960			

Table 4.1: Experiment data set statistics.

4.3.2 Crawling and audio pre-processing

The *Audio Set* data set is based on audio tracks of Youtube videos. All samples are already available as embedded from the *Audio Set* page. This original publication has the side effect that the data is only available already processed by a CNN called VGGish [39]. Using those embeddings allows to only build a classifier on top of an already trained model, but no actual processing of the raw data is possible. The VGGish network is available as Tensorflow v1 implementation and checkpoints [2], [82]. This is quite outdated and limited, so the embedding is not used for this work.
Instead, a crawler is created that downloads the audio tracks from Youtube. For this, the pytube3 library is used to use the latest publicly available Youtube version and extract the video and audio streams [25]. Using this tool, the crawler downloaded about 1.1 million audio clips. As the audio tracks on Youtube are in a compressed mp4 format, they are first transferred to wav format with a bit rate of $44.1kHz$. If the downloaded segments are too long or too short, they are corrected if possible or dropped. The 'wav' files are available for further processing.
There is a simple implementation within the librosa library [56] that concatenates the transfer to the frequency domain and creates a Mel spectrogram. As parameters, a hop length of 441 data points or 0.01 seconds, a Fourier transformation window of length 2048 data points, and 64 Mel bins are used. This results in a Mel spectrogram of the size 1001 x 64 used for the training. All spectrograms are pre-processed and stored for the activity.

4.3.3 Training of the teacher ensemble

The training approach for a *PATE*-based classifier consists of two main steps. First, The model creator must train the teachers, and afterward, the student model can be learned. To invoke the training process, a data set must exist (see section 4.3.1) containing samples for training and testing of all relevant models. The pipeline can start every training step individually if the previous actions are closed. So the student training requires the teacher training to be finalized before it can be kicked off, but it is possible to train different student models on the same teacher ensemble in parallel. This leads to the situation that the performance of the individual steps should be analyzed independently from each other as described in the evaluation section 5.
For all model training a common set of configuration properties influence the training or can be observed during the process. Multiple design-related and operational preferences must be evaluated for the model training. On the model itself, the model architecture as described in the section 4.4.1 and the number of classes it must predict have a high impact on the training.
Further, the creator of the models can monitor more operational attributes as learning rate, batch size, and the count of iterations in a tensorboard [2]. Those can be configured upon the run time of the training. These parameters and their impact on the training are further discussed in the evaluation section 5.4. Instead of epochs that describe that the whole training data is shown once during the training, iterations are used. An iteration represents passing a single batch within training through the model. This uncommon reporting technique is required for the feature batch balancing described later in the section 4.4.2 that shows samples according to their labels assignments and the probability of the label in the training set.
Training the classification models for the teachers is the most time-consuming task during the training process. As the number of teachers indirectly influences the privacy analysis, the aim is to have as many teachers as possible in the ensemble. On the downside, more teachers require the creation of more individual models. Even if the training of the teachers can be conducted in parallel, this step is the most time and resource-consuming for training a classifier with the *PATE* framework.
Compared to a single model approach, this blows up the effort to create a classifier by a factor close to the number of teachers involved. All Mel spectrograms are prepared or downloaded to train an individual teacher first. With the training parameters, then a model is trained for a pre-defined set of iterations. The checkpoint of the model is then saved for predicting the student samples later on.
After training the complete list of teachers for an experiment, the preparation for the student training can be started. As the trained ensemble can be reused to try different approaches for the student training, the goal is to make as much reusable as possible. With this in mind, the training set for the student gets predicted by each teacher and stored in fast-to-access array structures. Such arrays organized per sample or per teacher are stored in a file. These pre-calculated predictions are then used to train the student model. Even in a real scenario with teacher models distributed among different data owners, such a pre-processing step of the ensemble predictions must be performed. The ensemble prediction can not be publicly released as it contains private information from the data used to train the teacher models.
During the whole training process, the different models are tracked for performance characteristics. The standard validation set described in 4.1.2 allows comparing the performance of the

individual models. Classical metrics like precision, recall, and the combined F1 score enable to monitor the quality of the model. For multi-label classification mAP and AUC allow a objective view among all classes of the model. They concentrate the information of the precision-recall curves of all classes into a single value. In addition, metrics are reported class-wise. As discussed later in the evaluation section 5.1.1, the classes perform independently from a model or other training parameters differently.

4.3.4 Preparation of aggregated ensemble votes

The implementation of the aggregation mechanism is closely linked to the privacy analysis. For all theoretical explanations on aggregation and privacy, please refer to the respective section 4.2. Here the technical implementation of the preparation for the student training is laid out.
After all teacher models have been trained, the ensemble can perform the teacher votes' aggregation for the separate non-private set of student training data. The ensemble predicts this data and assigns labels to each sample. In this thesis, this step is independent of the actual student training. For a real-world scenario with data distributed among the data owners that do not trust each other, this step might be performed ad-hoc together with the privacy analysis and the student training. As input for the aggregation of the teacher models, the student training data, an ensemble threshold th_e, a confidence threshold th_c, and two noise parameters σ_e and σ_c are required. In case no confidence mechanism is applied the th_c and σ_c are not set. Those four parameters and the used aggregation mechanism characterize the teacher ensemble aggregation. The produced outputs can be used for various training of student models. The predictions of the teacher ensemble are stored in efficient data structures for re-usability. This allows later to evaluate various aggregation configurations quickly. Based on the accumulation of the ensemble votes metrics, the future performance and used privacy-cost of a student model can be estimated. The aggregated votes are the foundation for the student training.
The regular GNThreshold aggregation, the GNThreshold Adaptive, and the corresponding "confidence" versions efficiently aggregate the ensemble votes for the given parameters and predictions.
In parallel to the prediction, the privacy metrics become analyzed and reported. The probability q that the actual aggregation result will not be outputted due to the application of noise, the data-independent and data-dependent privacy parameters are calculated for each sample. As RDP allows to efficiently aggregate multiple for the same order, the cumulative privacy is also calculated in the RDP domain. This results in a tighter privacy guarantee than adding up the privacy cost in the (ϵ, δ)-DP domain due to composition theorems. All privacy metrics will be reported as ϵ values for (ϵ, δ)-DP with an $\delta = 10^{-8}$.
This step permits a comprehensive view of the performance of the ensemble. The ensemble's performance has a significant impact on the student performance later. Hence the ensemble performance is evaluated in section 5.2.

4.3.5 Student training and privacy analysis

Training a separate student model is the critical element to fixing the privacy cost in the *PATE* framework. All models are not aware of each other and have fully separated data. The label assignments of the student training created by the ensemble are transferred in a privacy-preserving

way to the student. The test set is shared among teachers, the ensemble, and the student to allow a comprehensive evaluation.
Even if it is theoretically possible to train each model with a different model and training approach, the configuration is not mixed within an experiment for simplicity. For the final student training, the files with the predictions of the teacher ensemble must be available. Together with an aggregation mechanism further described in the section 4.3.4 and the regular training parameters, the student model is trained the same way.
The student model learns from the predictions of the ensemble. It is characterized by the number of training samples used for the training and by the aggregation parameters. Based on these attributes a fixed privacy cost is calculated for the student model. The samples with the label predictions can be shown as often as required without impacting the privacy cost of the model training. This step of the training process is evaluated in section 5.4.

4.4 Features for Audio Set classification

The *Audio Set* is a challenging audio classification task. The characteristics of this data collection require robust machine learning architectures. Additional features try to support the training process by mitigating unwanted data properties. The theoretical part of these tools is described below while evaluated in the chapter 5.

4.4.1 Used machine learning models

Resnet18, Resnet34, VGG11, and VGG19 architectures are used for the audio classification task. The building blocks, e.g., the Basic Block (BB) for Resnets, are provided by the Pytorch and torchvision library [79], [37]. All nets usually process image data with three layers for the three primary colors. In the audio case, just a single dimension indicates the relative energy at a specific point in time.
The classification layers are custom built and contain multiple fully connected layers, rectified linear unit (ReLu) activation layers, batch normalization, and dropout layers. For all of them, Pytorch standard implementations are used [69]. The last fully-connected layer maps the output to the number of classes. The impact of those layers is further discussed in the section 5.1.2.
For all those models, pre-trained snapshots are trained on large-scale image tasks. Even if this work does not intend to classify images, those pre-trained weights are helpful as standard patterns like edges must not be learned from scratch. Anyway, the parameters of the convolutional layers are not fixed during the training to let the overall model adapt to the audio classification task. This approach can be described as transfer learning from pre-trained models even if the initial training originates in a different domain.
The Adam optimizer is combined with the already outlined BCE loss for the backpropagation step [44].

4.4.2 Batch balancing data sampling strategy

Another approach to address the imbalance of the data set is to adjust the seed strategy for the training process. It defines in which order the samples are aggregated into batches. A batch

Resnet18	**Resnet34**	**VGG11**	**VGG19**
Log Mel spectrogram 1001 x 64 Mel bins			
(7x7 @ 64, BN, ReLu)		(2x2 @ 64, ReLu)	(2x2 @ 64, ReLu) x 2
Pooling 3 x 3		Pooling 2 x 2	
(BB @ 64) x 2	(BB @ 64) x 3	(2x2 @ 128, ReLu)	(2x2 @ 128, ReLu) x 2
Downsampling / Pooling		Pooling 2 x 2	
(BB @ 128) x 2	(BB @ 128) x 4	(2x2 @ 256, ReLu) x 2	(2x2 @ 256, ReLu) x 4
Downsampling / Pooling		Pooling 2 x 2	
(BB @ 256) x 2	(BB @ 256) x 6	(2x2 @ 512, ReLu) x 2	(2x2 @ 512, ReLu) x 4
Downsampling / Pooling		Pooling 2 x 2	
(BB @ 512) x 2	(BB @ 512) x 3	(2x2 @ 512, ReLu) x 2	(2x2 @ 512, ReLu) x 4
BB / Pooling		Pooling 2 x 2	
FC 512, ReLu, BN, Dropout@0.5		FC 4096, ReLu, Dropout@0.5	
FC 256, ReLu, BN, Dropout@0.5		FC 4096, ReLu, Dropout@0.5	
FC n		FC n	

Table 4.2: Model parameters of the used CNN architectures.

consists of a limited set of samples passed together through the model. Over this set of predictions, the model adjustments are performed.

For a good training result, the algorithm must balance a big enough collection of batched according to the class distribution. While the classes are not equally distributed, all classes should be present within a series of batches that is significant to the total amount of samples. If there are no specific requirements, the training set are shuffled to create a random order. This is then used to slice the batches for the training. This works well with data sets that have a close to uniform distribution among classes.

For the highly imbalanced *Audio Set*, this causes issues. The problem is the imbalanced distribution of labels. In the original data set, the second most class, “Speech” has one million videos assigned while the seventh most class, “Car” only has 42.000 samples, which means that there are 23 times more samples of “Speech” over “Car”. This could lead to situations in that batches only contain instances of the “Speech” class which impacts the training process in a negative way.

To address this problem, a custom seed process was introduced to the models’ training. Technically this is handled as a new implementation for the “Sample” interface of the Pytorch framework. All samples are organized by their assigned labels in an ordered array list “id_by_label”. Another list, “next_label”, contains all possible labels. Both lists are randomly shuffled internally. This sampler is used as an iterator that yields one video id after another for the composition of a batch. It pops the first item of the “next_label” list to identify the main label that the next sample in the batch should have. The first item is popped from the corresponding “id_by_label” list for this class. If any of the lists run out of items, they are reinitialized with their initial content and randomly shuffled again.

This approach does not guarantee equal distribution of labels among the batches. Due to multiple label assignments to each video id, there is still an imbalance to the more frequent tags. But it ensures that seldom classes appear at least once every few iterations. On the contrary, the training process might never show some samples of the more frequent labels during the training process. At least they are presented less often than for a less frequent class that has

its list of video ids refreshed frequently. This could result in overfitting to those few samples. The evaluation section 5.1.3 will discuss if batch balancing helps a classifier with the *Audio Set* collection.

4.4.3 Mixup data augmentation strategy

Most machine learning models perform poorly due to the limited availability of training data. One way around this is to reuse the existing data and create augmented versions of it. Data augmentation is a common technique to increase the data set's size artificially. It makes augmented samples based on existing ones by slightly changing them.
Algorithms can achieve this in a visual image learning model by cropping, tilting, or other operations on image data. For audio data, this becomes a bit more complex. A machine learning training can perform such changes in the time and the frequency domain. The model learns complex patterns from the pre-processed log Mel diagram images. Operations like tilting or random cropping would rather destroy the data than help the algorithm. Techniques for audio data augmentation must respect the nature of audio data. Basic operations like shifting in the time domain or scaling in the frequency domain should not eradicate the original information. At the same time, a shift in the frequency domain can significantly harm the data.
No augmentation on the raw data in the time or the frequency domain was performed in this work . Instead, data augmentation got applied on the pre-processed log Mel diagrams directly. For this, some samples got mixed with other samples. This creates a diverse representation of two examples to create a new sample artificially. This technique is referred to as "micup". The data loader for the training loop combines the label assignments with the same approach. This idea for the augmentation of data originates from [47]. The paper itself is one of the most promising approaches to improving the *Audio Set* data classification.
Two random samples of the training set are taken and mixed for the mixup approach. This approach can be understood as combining two sound events. Two input audio clips x_1 nd x_2 with their corresponding binary label vectors y_1 and y_2 are linearly combined. They are aggregated by $x = \beta x_1 + (1 - \beta)x_2$ and $y = \beta y_1 + (1 - \beta)y_2$ on the log Mel spectrogram.
This results in a linear combination of the two sound events in the frequency domain. This should work fine as it linearly adds up the detected frequencies in the samples. The factor β is sampled from a β-distribution that ranges from 0 to 1.
The size of the randomly arranged data set is increased by two using the mixup approach. For this, the x_i sample got combined with the x_{i+1} sample of this ordered list. The rate of presence of the label got scaled accordingly. The label is assumed to be present for batch balancing and further analysis even if just a value slightly greater than 0 is assigned to y.

4.4.4 Loss function class-weight scaling

One big issue of a hierarchical multi-label data set is the high imbalance of the classes. As described in the section 4.1, every sample can have multiple labels assigned. If the data set creators were unsure about the correct sub-class, the given label got the common parent class. The other way around, not all parent classes of an assigned label are necessarily mapped too.
This leads to the situation that classes on the highest hierarchy level as "music" or "speech" have more than a million samples while more specific lower-level labels have less than 10.000 samples.

This can highly impact a machine learning model. If a label is assigned to a high portion of the overall samples, it can lead to a situation where the model consistently predicts this class. This happens if the penalty for predicting the class becomes less than the penalty for always predicting it. The other way around, if it is best not to output a class label in most cases, the model can falsely always not label this class. Both extremes are unwanted outcomes for the training process.
As a criterion or loss function, the BCEWithLogitsLoss combines a sigmoid layer with a regular binary cross-entropy. It similarly encourages a classifier to treat positive and negative classifications in a multi-label scenario. The mechanism adds the logarithmic probability that the class is predicted correctly according to the actual values. This sum is then averaged over all classes. Adding weights to the base loss function defined in 1 this formula makes it possible to trade off precision and recall per class. This can prevent that the trained model tends to one of the extreme situations described above.

Definition 13 (BCE weighted loss)**.** *The weighted BCE loss is defined below with t as binary target value and y as model output. A weight factor $p \in \mathbb{R}^{|t|}$ is added to the positive outcomes.*

$$Loss = -\frac{1}{|t|}\sum_{i=1}^{|t|} \boldsymbol{p}_i t_i \cdot \log y_i + (1 - t_i) \cdot log(1 - y_i) \tag{4.9}$$

The weight per class is chosen according to the overall number of samples in the data set. A common approach is to divide the negative examples for a class by the positive ones. This improves the recall for the seldom classes while harming the precision for this class. This simple calculation for the weight tends to extreme values for the weight. The *Audio Set* data can explode quickly in values greater than 10.000. Instead, the fraction is passed into a sigmoid function to bind the weights by a maximum and minimum value. This thesis will further discuss the benefits of those approaches in the evaluation section 5.1.3.

5 Evaluation of an Audio classifier for city and environmental sound

In this section, the described implementation approaches of section 4 are evaluated. For this, the metrics defined in the section 2.1.2 are used. Within the *PATE* framework, it is easy to measure the performance at different stages of the framework. Each stage produces a model that can be tested independently. The overall performance of the private student model, in the end, builds upon the intermediate model capabilities.
For all evaluations, the official evaluation set of the *Audio Set* is used, which is further described in the section 4.1. Only the evaluation samples containing at least one label considered for the model's training are used. It has at least 50 examples per class, while it is not evenly distributed due to the assignment of multiple labels to a single sample. This data set theoretically allows a comparison to other works on the *Audio Set*. Unfortunately, all metrics are averaged among all individual classes. Here only sound categories related to city and environmental sound events are used. As demonstrated later, each class performs differently. Hence, the classes' selection has a high impact on the reported overall performance. This aspect must be considered while comparing to other works.
The below sections focus first on the class-wise performance in section 5.1.1, individual teachers and their technical model parameters in sections 5.1.2 and 5.1.3. This concludes in the teacher ensemble evaluation in section 5.2 and the student performance in section 5.4. The last topic is privacy analysis in section 5.3, which is the critical element of this work before concluding. The performance of the individual steps is essential even for the privacy analysis, as the more aligned the ensemble is, the more private the mechanism can become.

5.1 Teacher Performance

PATE is a federated learning approach. The single teacher is the critical component for the performance of the model. A teacher can be an independent data owner who trains a private model. Every teacher component contributes to the general learning task in such a setting. If a single teacher has strong prediction skills, the aggregated model profits from that. For *PATE* additionally, the privacy consumption depends on the consensus of the ensemble of the teachers. If every teacher predicts a sample correctly, the privacy cost is close to zero. The prediction performance of every teacher model is therefore essential for a solid and private classifier under the *PATE* framework.
The available training data is the limiting factor in most machine learning tasks. A federated setting can solve this problem as nobody has to share private data with others. But training many different individual models on small portions of the overall data can be challenging as well. The individual models become useless if the task is too complex to be trained on just a tiny subset of the data. Even if the assumption is that the aggregated classifier outperforms the individual ones.
For the *Audio Set*, the overall set of one million data samples are split into random subsets.

Those random subsets have similar properties as the general data set. For example, they are highly imbalanced. In the *PATE* framework, the number of teachers has a direct impact on the exposure of private information by the aggregation. This can be imagined as private information can be hidden behind the large ensemble. If many parties contribute to the result, an individual teacher's impact becomes limited.
As the *Audio Set* is not infinitely huge, the number of teachers must be balanced with the count of the training samples used for a single teacher. No model knowledge sharing techniques can be used across the teachers with the *PATE* framework as this would destroy the privacy guarantees. The challenge for the training of individual teachers is to train them with a minimal number of samples. Instead of achieving the overall best performance with unlimited training data, the aim is to make the model as good as possible with limited training data. In the case of the "big 20" experiment set, each teacher is trained on about 14.000 samples. This is quite challenging for a multi-label classifier with more than 150 classes.
For the following subsections, different types of evaluations are used. Single value metrics can be observed by the number of iterations performed in training. An iteration refers to a single training loop. Within such a loop, the batch samples are shown to the model. Based on their predictions, a backpropagation step is performed. The iteration multiplied by the batch size determines the number of training samples delivered to the model. Hence, the number of examples shown must be considered when comparing different batch size configurations. All training samples will be reused in the training loop. As some classes have just a small number of samples, this replacement is mandatory.

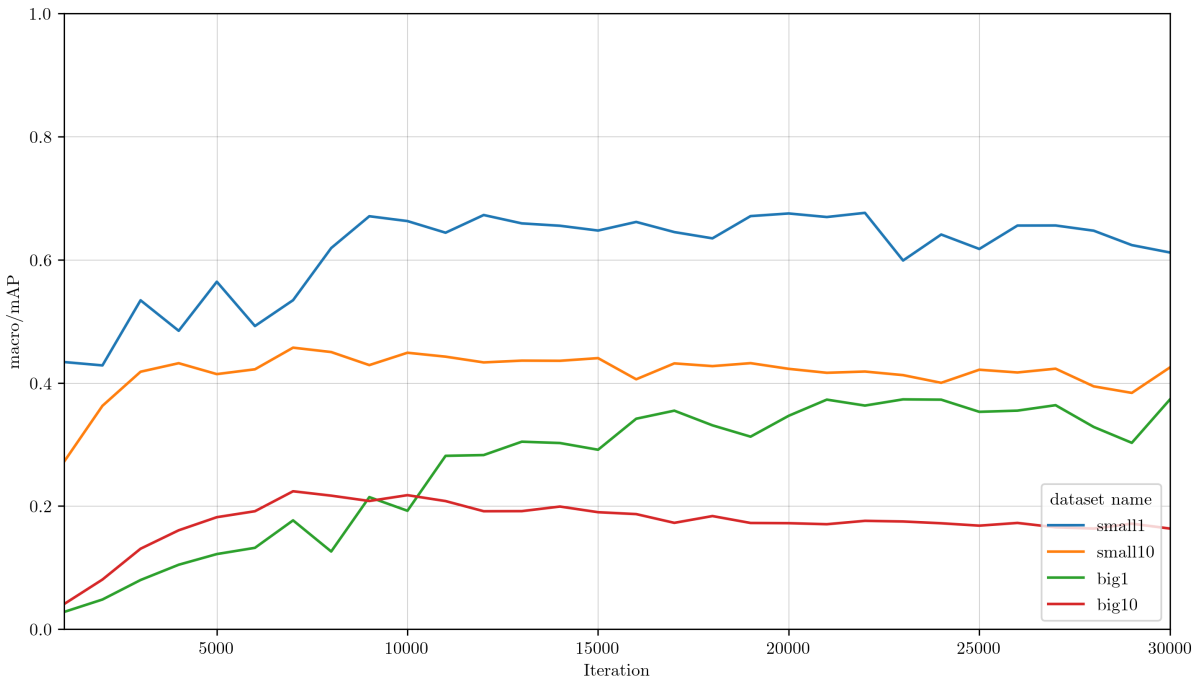


Figure 5.1: Performance of individual teachers over iteration.

Typical single value metrics are mAP and the averaged AUC, described in the section 2.1.3. Both metrics are calculated by class and then are averaged to have a single value description of the performance. They can be observed relative to the training progress of an individual teacher.

The figure 5.1 shows the mAP for teachers of the different data sets over the training iterations. For this plot, the optimal configuration of the model and model parameters are selected. Only classifiers of the same category, “small” or “big”, can be compared. Otherwise they have a significant difference in the output classes and, therefore, in the task’s complexity. For the initial discussion, they are reported together. Later on, they will be observed separately.
The less complex “small” classifiers with only 15 classes perform better than the “big” classifiers with 153 classes. But even the classifier with the limited number of classes is not close to a mAP of 1.0, which corresponds to an optimal precision-recall curve. This matches with other works on the *Audio Set* that report similar results. The reason for this result is further discussed in the section 5.1.1.
The ensembles with only one teacher serve as the comparison baseline for the private and aggregated models later on. These models are trained on all available data without any privacy-preserving techniques. They will be referred to as “baseline” in the following sections. The “small” classifier can achieve a mAP score of about 0.65 in a non-private scenario, while the “big” model only achieves a mAP score of about 0.35.
All classifiers learn steadily in the first 10.000 iterations. Afterward, the performance differs by the models. While the single teacher models with the maximum amount of training samples mainly stagnate or slightly increase in their performance, the teachers with reduced training samples decrease. This goes back to over-fitting in the training process due to a small number of training examples. As shown later in the section 5.1.3, adapting the learning rate or batch size can not fully solve this issue. Training a multi-label classifier on complex audio data for about 150 classes requires a sufficiently large training set to not over-fit. Since the *PATE* approach requires many teachers and the data is limited, the training is stopped after a decrease in performance of the teacher is detected.

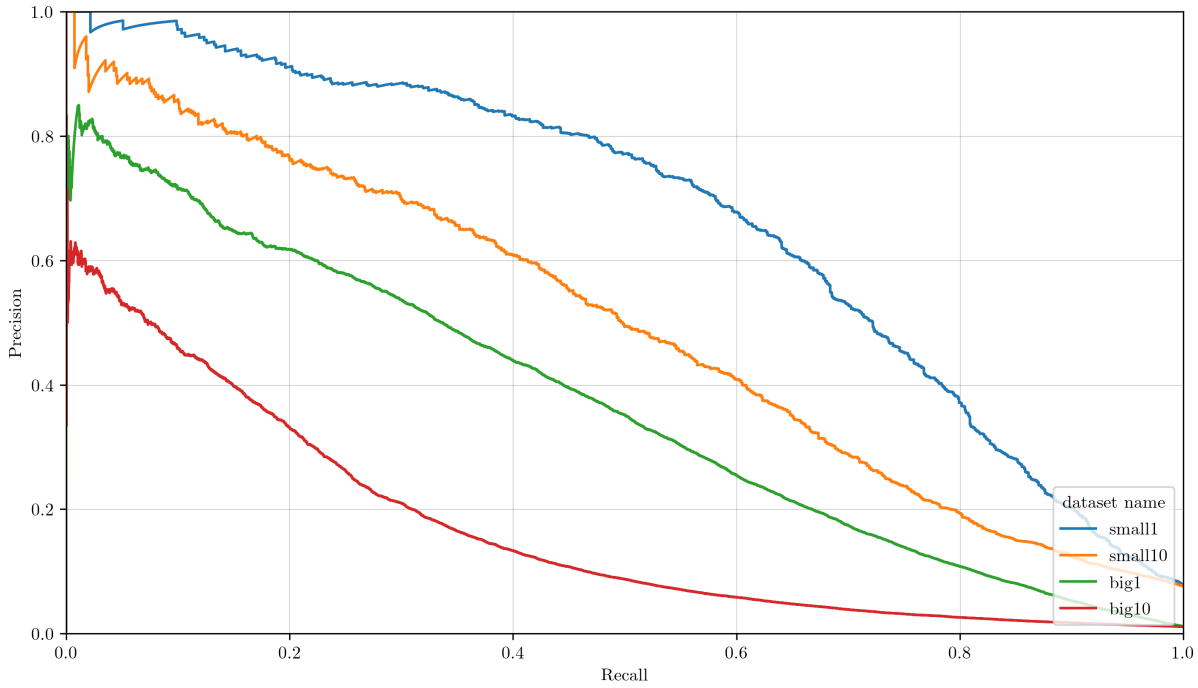


Figure 5.2: Precision-recall curve of individual teachers after iteration 30.000.

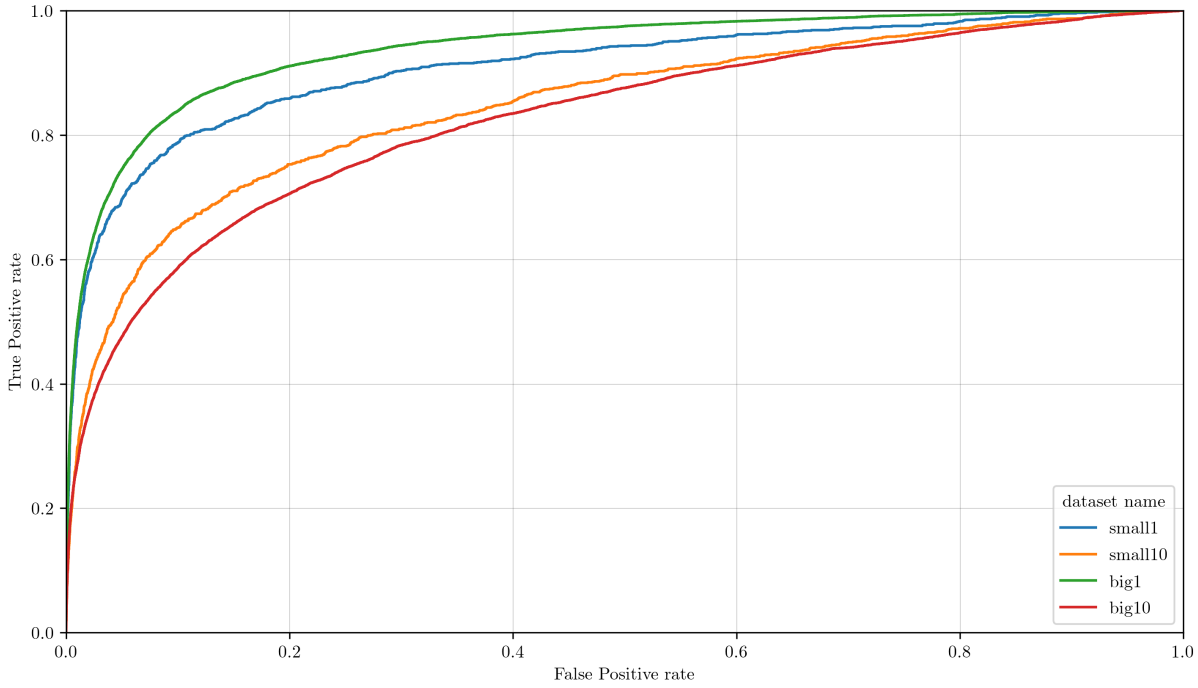


Figure 5.3: ROC of individual teachers after iteration 30.000.

The figure 5.2 shows the precision-recall curve, and the figure 5.3 shows the ROC curve for the discussed models after 30.000 iterations of training. The precision-recall curves correlate to the mAP, which reports the area under this curve. The difference between the performance of the different training sets is significant.

Just observing the precision-recall curve could lead to the idea that the models with limited training data have no skill to predict at all. The peak of the curve is somewhat in the bottom-left corner instead of the optimal top-right corner. In such cases, the ROC curve helps to understand if the training can retrieve any knowledge from the data. As discussed in the section 2.1.3 it reports the ratio of true-positive predictions over false-positive predictions among different thresholds. This second diagram shows that if the model predicts something, there is a good chance that it is correct. All ROC curves have a solid peak to the top-left corner of the diagram, which represents a skillful model. Interestingly, the “big” classification model outperforms the “small” one for the baseline. This might relate to the composition of solid and weak performing classes, as discussed in the section 5.1.1.

The main issue of the *Audio Set* seems to be to detect labels at all. The recall suffers from the sparse label assignments as discussed in section 4.1.2 which lead to weak predictions and even partially incorrect predictions. For the training of an ensemble, an poor recall is challenging. Only if the ensemble strongly agrees, it can derive precise predictions, and the privacy consumption is low. On the contrary federated learning can be an approach to mitigate poor predictions. Every individual teacher becomes an expert on a specific problem. The aggregation of the individual teacher predictions is further discussed in the section 5.2.

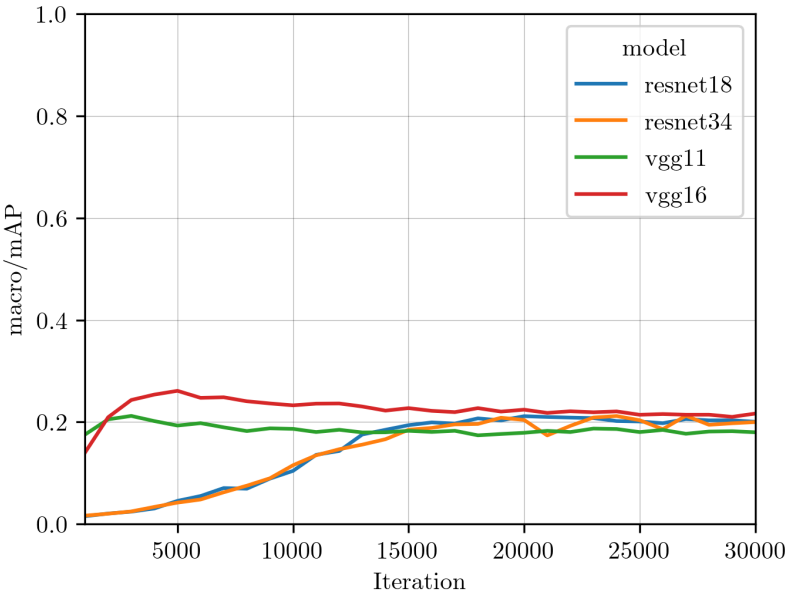
5.1.1 Individual class performance

Various metrics can be evaluated to visualize the multi-label classifier performance for a teacher. On a high level, all performance metrics for multi-label classifiers can be split into single value metrics that describe the whole classifier and class-dependent metrics that address the different classes individually. Single values are easier to understand and compare, but in a multi-label setting, the performance of individual classes can differ significantly.
This applies especially to the used experimental data of the *Audio Set*. The labels in the data are not only on a hierarchy that represents different levels of granularity, but they hold various difficulties inherently. Sounds like a "siren" have a unique pattern in their common understanding and the log Mel representation, while labels like "bus" are ambiguous. There is no clear sound pattern that appear in a persons mind hearing this label.
In the figures A.2 and A.3 that can be found in the appendix, the average precision by class is presented over iterations. The three plots show the ten best, medium, and worst-performing classes. The already discussed mAP represents the average of all graphs for all labels. For the figure A.2, all available training data is used, while in figure A.3 the training data is split among ten teachers.
It can be observed that the best performing classes show a strong similarity among the categories. Classes like "church bell" and "fire alarm" offer an excellent performance independent from the number of samples which supports the idea that some classes are easy to learn. On the contrary, classes as "rustle" and "bicycle" perform equally badly independent of training data. The classes with bad performance are either hard to imagine or usually appear as a secondary sound that is not noticed next to the main sound event. For a "bicycle," the sound can originate from the bell, the gears, the tires, or any other action. But there is no clear idea of how a "bicycle" sounds. "Rustle", "Crackle" or "Rattle" tend to be background noise. This is hard to distinguish and specify without any visual data in addition to the audio snippet.
With all possible training data available, the peak performance for the best performing classes is an average precision of about 0.8, which is close to an optimal score. This decreases together with the number of training data for a teacher. If only a tenth of the training data is available, the best performing classes gain an average precision of about 0.6, a decrease of 25%. This is even more significant for the worst-performing classes. While for the "big baseline" data set, at least an average precision of 0.1 is achieved for every class, the worst-performing classes with the smaller training set are close to zero. It seems like no skill at all can be learned by the teacher model for these classes. This might lead to a problem for the ensemble as well. If too few teachers show a meaningful skill for a class, the aggregated prediction and guaranteed privacy must be weak.
Similar results can be observed in the ROC curves in figures,A.4 and A.5 that are placed in the appendix. They present the same configurations and classes. The left plot with the ten best classes shows a solid skill for these labels. For the total training data, even the medium and worse performing classes offer a clear bump in the top-left direction for all classes. Only the worst class graphs for the "big 10" training data set are close to the diagonal, a no-skill classifier. How this impacts the whole teacher ensemble will be further discussed in the section 5.2.
It could be the case that the performance of a class directly correlates to the number of samples available during the training. Classes with a high number of available samples for the training

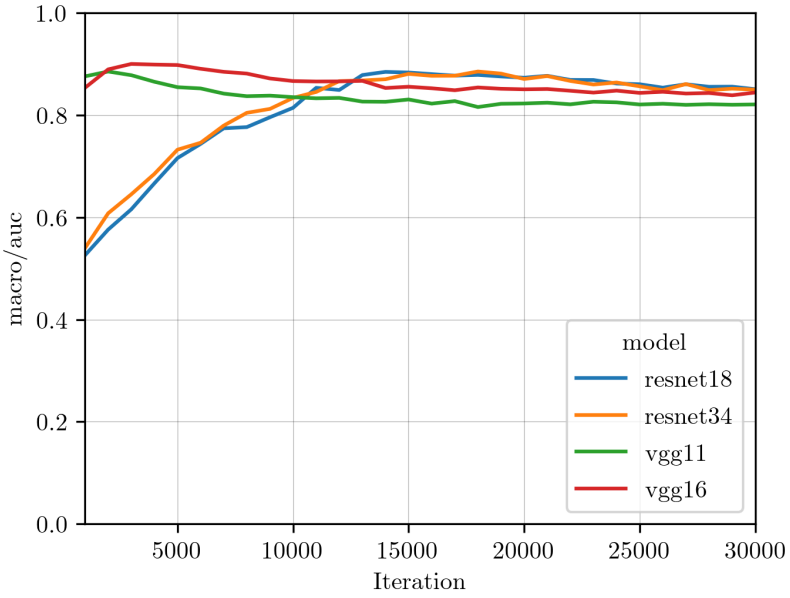
perform better and the other way around. In figure A.1, all 153 classes are reported with their average precision score and the number of training samples available during training. This graphic is reported only for the "big baseline" training set to avoid any side-effects of the randomly created teacher training data distributions. The classes are ordered decreasing by their average precision score. The blue bars report the total number of training samples on a log scale, while the red marks show the average precision. There seems to be no correlation between the number of available training data and a class's actual performance. For example, a class "Boing" performs about seven times better than "male speech" with a hundred times fewer training samples available. This confirms the assumption that some classes are easier to learn as the sound event is more clearly distinguishable than others.
The critical result of this section is that not all classes perform similarly. The considered classes have a significant impact on the overall performance of the individual model and later the ensemble. This must be regarded while comparing to other work that used all classes or another subset of the *Audio Set*. In addition to the single value metrics, the distribution of the average precision among the classes has to be considered.

5.1.2 Model performance and comparison

A machine learning model is a set of operations and layers. During the training the model become optimized to represent the data in the best way. Those operations are connected by weighted links between the layers and operators. In the weights of the inner layers, the model stores the prediction knowledge. Different model structures, sizes, and architectures lead to a different outcome and prediction quality. This section aims to elaborate on existing models and check their appropriateness for the classification of audio snippets. No new models or technical improvements will be worked on. Even if the used models are not relevant for the *PATE* framework directly as it is model agnostic, they influence the overall performance and privacy. The better the model supports the task, the better the overall prediction performance and privacy are.
The input data of the *Audio Set* are small images representing the log Mel diagram. They are fed as 1001x64 sized images to the models during the training. The long side represents the temporal component of the audio clip (10 seconds), and the short dimension represents the 64 Mel bins. With such images, most research and concepts for image classification apply to this task. In the following different deep neuronal networks are compared for their performance on the *Audio Set*. It is further verified if transfer learning from another domain with actual image tasks improves the audio classification task. Therefore the pre-trained models of the Torchvision library [54] are used that trained the models on the imagenet data [15].



(a) mAP score



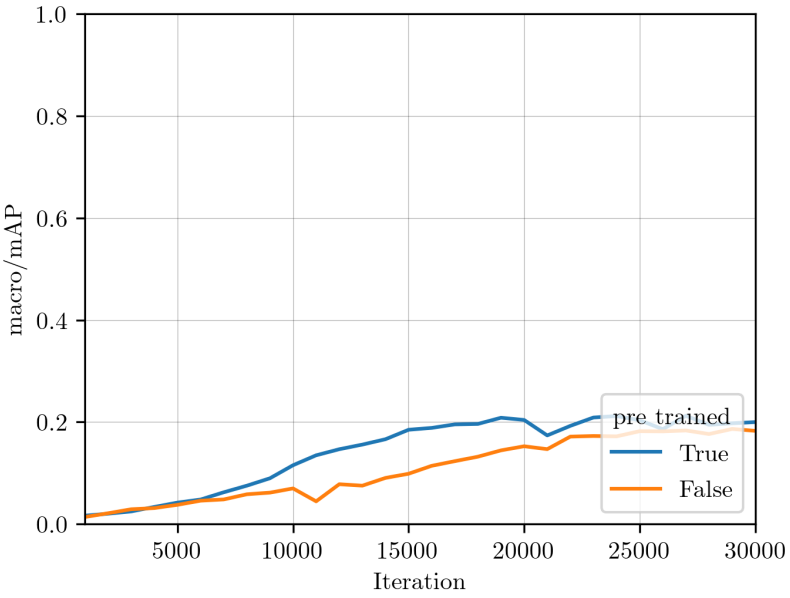
(b) AUC score

Figure 5.4: Teacher performance for different models for the "big 10" experiment set.

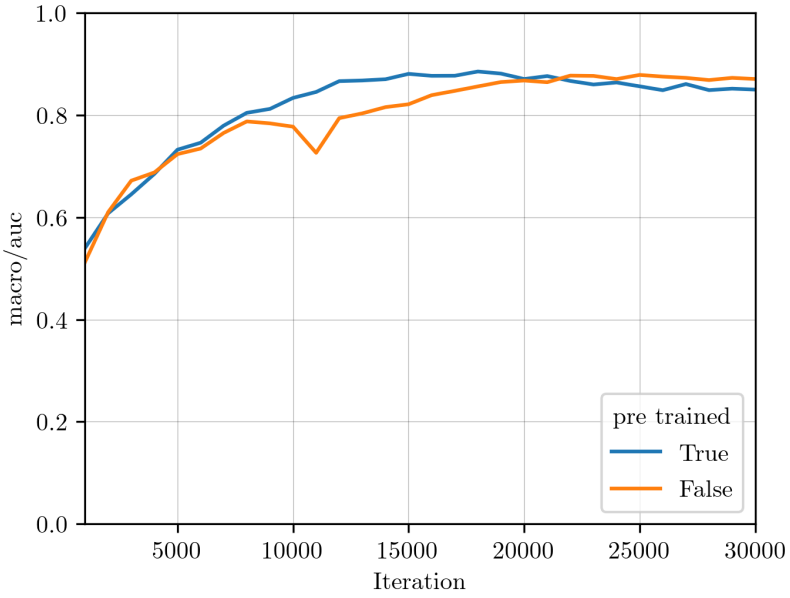
The figure 5.4 compares the different model types. Standard model types such as VGG and Resnets are used for the evaluation, as discussed further in the section 4.4.1. They only differ slightly in the fully connected layer to allow different sizes of output classes. All models are trained on the same data with the best standard configuration and hyperparameters. The main differences are the model architecture itself.

It can be easily seen that VGGs learn from the data much faster while having a lower performance in the end. This might go back to the vanishing gradient problem as these nets have about ten times the parameters of a comparable Resnet. They instead gain the knowledge slower but are more steady in the end. The exceeding performance of the VGG nets might result from the pre-trained model parameters as discussed below. The decrease of the prediction performance

derives again from the limited available training data for the teacher within an ensemble. For the VGG the AUC is worse, while the overall mAP is slightly better, points that the VGG variants have a better recall. The AUC of the ROC curve describes how skillful a model is if there is something predicted. Combined with a higher mAP score, this is a sign of stronger recall quality. As expected, more giant nets perform better as they are capable of storing complex structures. Detecting sound events from a single log-Mel diagram out of hundreds of classes is not a simple task. For the VGG variants, the better performance of the larger variant is visible. Instead, the Resnets show just a tiny improvement over the smaller model of the family.
The thesis mainly uses Resnets as their training time is up to five times lower than for a comparable VGG net. This primarily originates from the additional parameters in the model that require additional calculation for the gradients. Especially the ensemble training with a lot of individual models benefits from the shortened training time. The performance loss compared to this is relatively insignificant.



(a) mAP score



(b) AUC score

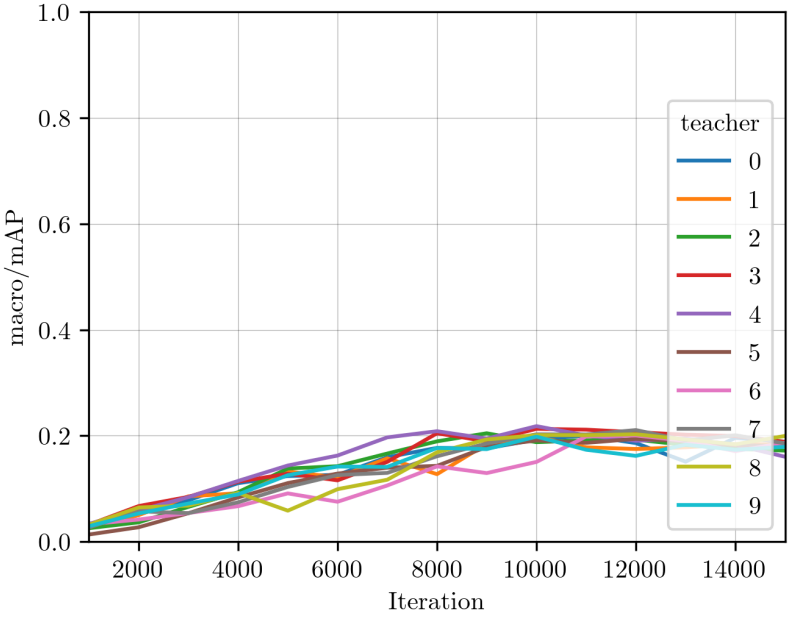
Figure 5.5: Teacher performance with transfer learning and without over iterations for the "big 10" experiment set.

All models come shipped with pre-trained parameters. These reference models are trained on huge image data libraries. As this is a different domain, it is interesting to see if the pre-trained configurations improve the results. In the figure 5.5, a single teacher is trained for the same data and training parameters with and without the pre-trained model.
The model that uses transfer learning outperforms the model with random initial parameters. With transfer learning, reusable features as edge or pattern detection do not have to be learned from scratch. This is very valuable for training with minimal training samples. Interesting is the behavior of the AUC scores beyond 25.000 iterations. It shows that the model trained only on the audio data offers a better true-positive over false-positive rate. But as the main concern

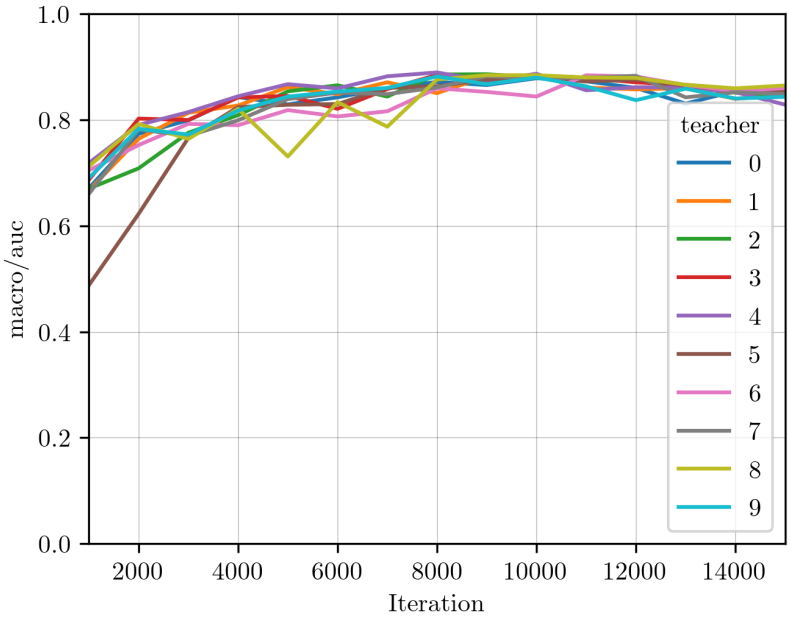
for the ensemble training is to let the models detect weak performing classes at all, the better average precision among all recall levels is the preferred option. Additionally, transfer learning allows faster model training and earlier stopping the training process. Both are valuable assets if training a high number of teacher models is required.
This section shows that the selection of a model and especially transfer learning can impact the overall model performance. While VGGs might be more powerful in a perfect scenario with unlimited training data, the training time advantages of Resnets make them the preferred option for this work. Transfer learning is a helpful technique to mitigate the lack of large training sets.

5.1.3 Individual teacher performance

The individual teacher is quite essential for the overall performance of the ensemble. As every teacher is trained on a small subset of the data, they often lack performance. If the teachers make ambiguous decisions, the ensemble can not confidently vote on a sample. This leads to weak overall performance and high privacy cost. Hence, tuning the individual teachers' predictions is one of the essential activities for the training of a *PATE* ensemble. For this purpose, different mechanisms are observed that improve the prediction quality significantly. The biggest issue to overcome is the small amount of training data used for the teachers. While it is not expected by the individual teacher to have the performance as the whole ensemble, it must learn to predict as well as possible.
In the below figures and discussion, different features and parameters for the training are discussed. With hyper-parameter tuning tools the optimal values were collected for the final evaluations. Here the parameters and options are discussed isolated to analyze their impact on the classifier training.



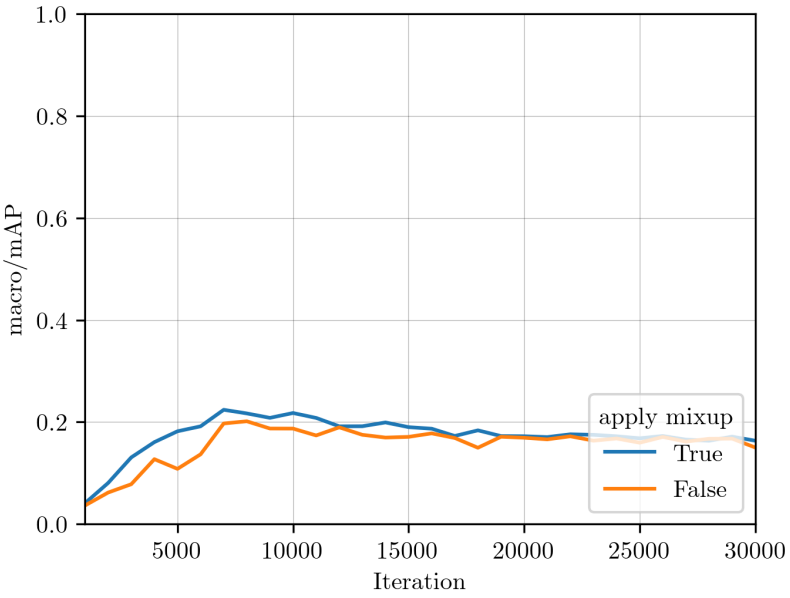
(a) mAP score



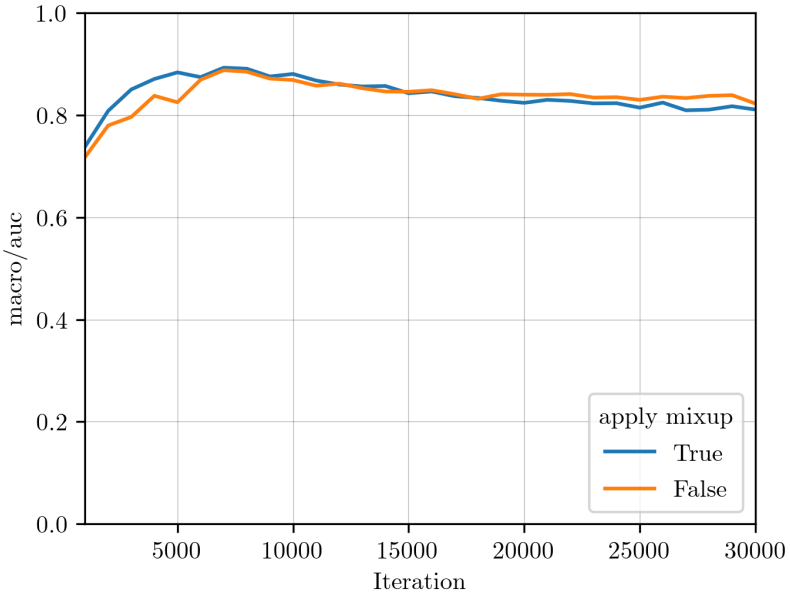
(b) AUC score

Figure 5.6: Performance of all ten teachers over iterations for the “big 10” experiment set.

The following features are always discussed based on teachers trained on the “big 10” experiment data. This is close to a real scenario for later ensemble evaluation as the teacher usually does not have all training samples available. It is worth analyzing if the teachers trained on a comparable set of training samples have a similar performance. In figure 5.6 the mAP and the AUC for the ROC curve are plotted. It can be seen that all teachers of the possible ensemble have similar learning progress and maximum scores for the two values. There is no significant difference between the teachers if the training parameters remain the same. For the rest of the work only a representative teacher is reported to keep the following plots clean.



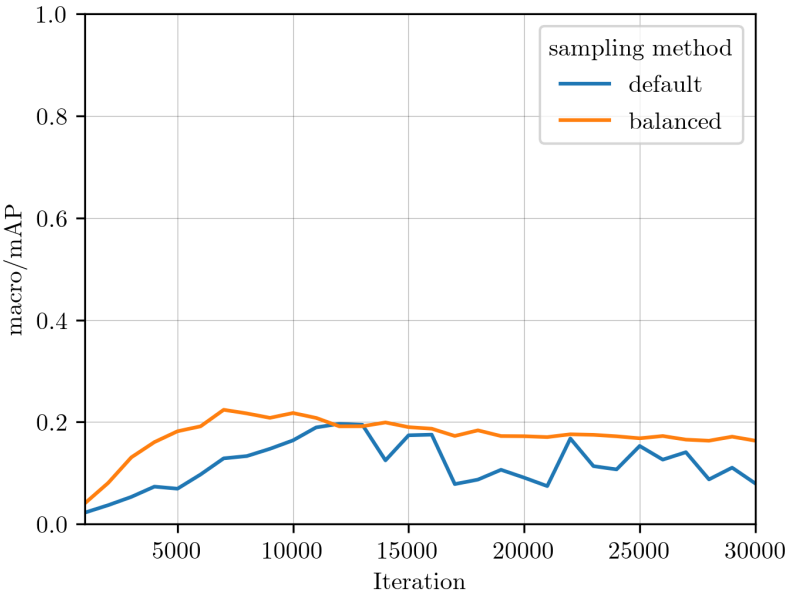
(a) mAP score



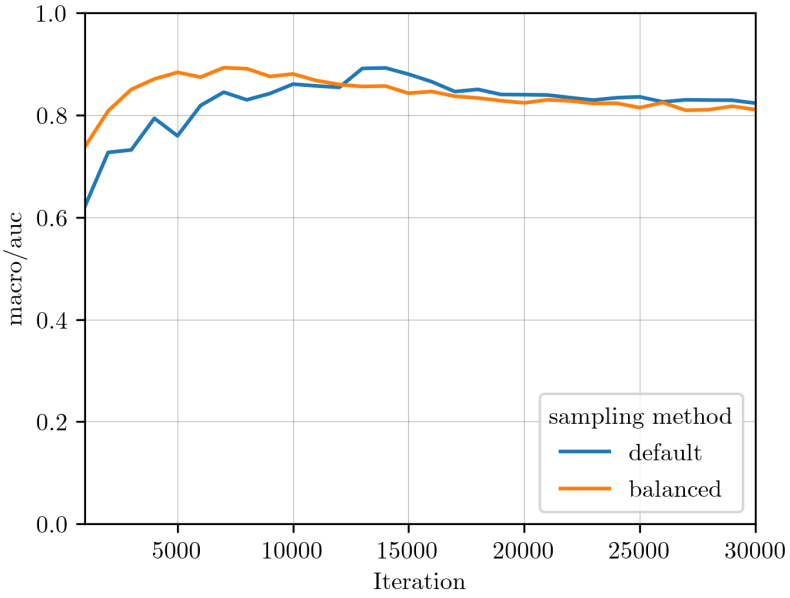
(b) AUC score

Figure 5.7: Performance of the "big 10" data set with and without mixup over iterations.

The described mixup technique of section 4.4.3 allows increasing the training data artificially. For this, two samples are linearly combined to a new combined sample. A slight improvement can be seen in the figure 5.7 if the mixup technique is applied to the training process. Especially in the early stages of the training, mixup helps to learn faster. This fits the problem that the *Audio Set* data is only weakly labeled, which makes it hard for the classifier to deal with multiple or ambiguous classes on a sample. Mixup creates additional examples and introduces partial samples with reduced intensity. This has a positive effect on the training process. As it allows potentially higher peak performance and stabilizes the training process, this thesis applies mixup data augmentation for all reported evaluation.



(a) mAP score



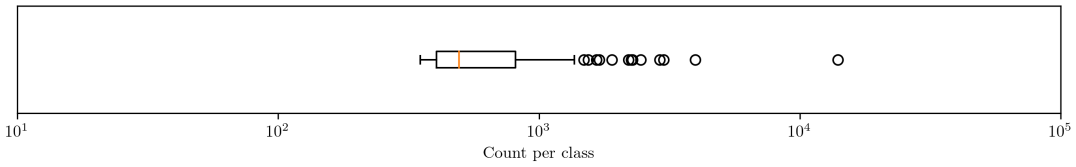
(b) AUC score

Figure 5.8: Performance of the "big 10" data set with balanced and unbalanced sampling strategy over iterations.

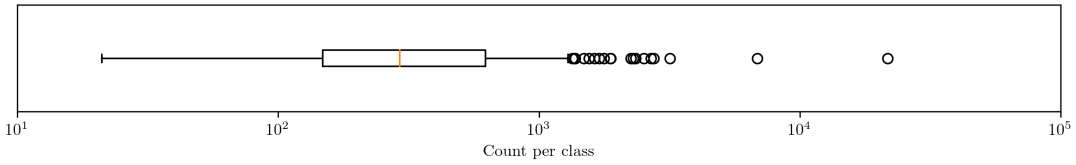
In section 4.1.2 the imbalance of the *Audio Set* data is discussed as one of the major problems. This imbalance can be seen while observing the class distribution of batches for the default sampling strategy. Instead the balanced sampling strategy select samples according to their class assignments. This creates a better balance between the labels within the batches (see section 4.4.2).

In the figure 5.8 the effect of batch balancing can be observed. The classifier learns much slower with the random sampling technique, and the performance scores oscillate heavily. Even the overall performance benefit from the balancing. Having an equal distribution of labels per batch stabilizes the training.

For the AUC score, it can be observed that the balanced sampling performs slightly worse than the default sampling. Together with the performance of the mAP score, this must be related to a few classes that are favored by the classifier. They always have a significant share in the batches, and therefore the model learns to predict them more often. If this happens and the model is correct as they have a high frequency in the test set, the true-positive rate increases, but the overall performance for other classes suffers. Controlling the distribution of classes in the batch of samples seems to be a powerful strategy to improve the training on the *Audio Set*.



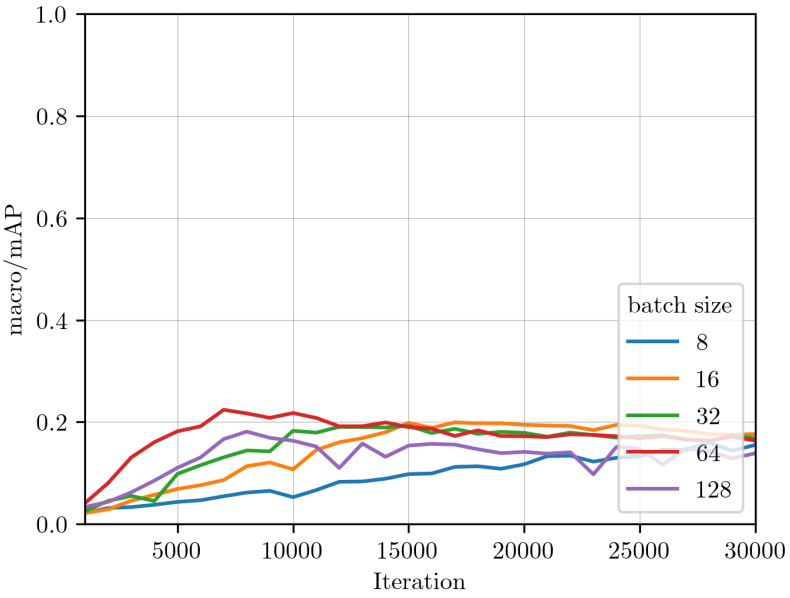
(a) balanced



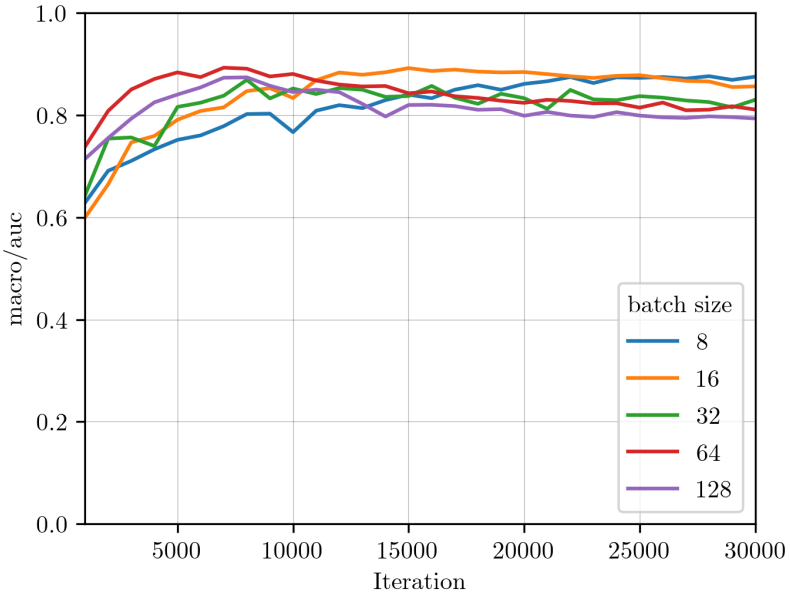
(b) unbalanced

Figure 5.9: Distribution of classes in 1000 iterations with balanced and unbalanced sampling strategy of “big 10” data set on a log scale.

The figure 5.9 illustrates the effect of batch balancing. It shows the distribution of the occurrences of classes in 1000 iterations on a log scale. For this, the number of class representatives in a batch is summed over 1000 iterations. The distribution among the classes becomes tighter compared to random sampling. This means that the different classes are more likely to appear in a batch. Regardless there is still a strong outlier. The characteristic that causes this is that the parent nodes of the ontology hierarchy are more likely to be present than the more specialized classes. The samples have usually more generic labels assigned additionally in most cases. The most right outlier point corresponds to the data’s maximum available class “vehicle”.



(a) mAP score



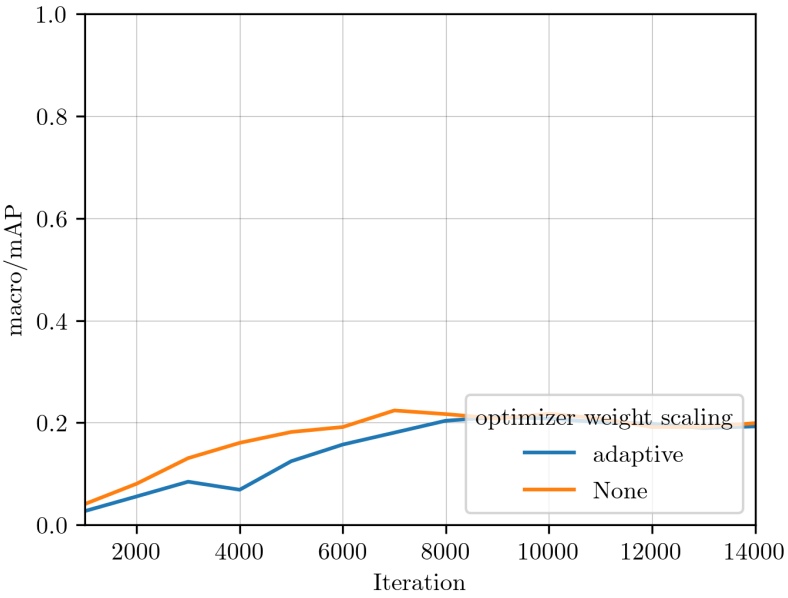
(b) AUC score

Figure 5.10: Performance of "big 10" data set for different training batch sizes over iterations.

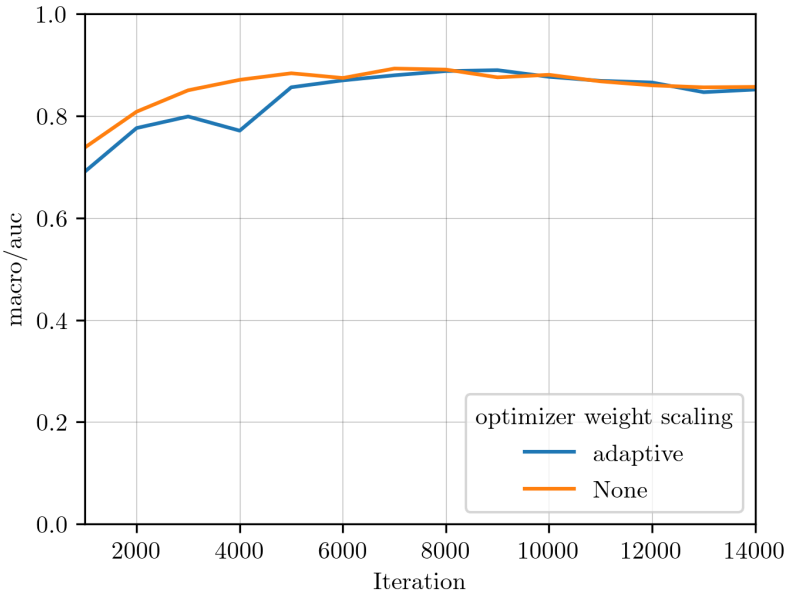
The batch size defines how many samples are shown to a model in the training process. Over the whole batch, the loss is calculated, and a backpropagation step is performed that adapts the inner weights of the model. The best value for the batch size allows a model to generalize while still learning from a particular batch configuration properly. If the batch size is too small, a model learns the individual samples and overfit them. The model can not discover any specifics for a too-large batch size as the calculated loss is always the same.
In the figure, 5.10 different batch size configurations are shown over training iterations. Smaller batch sizes let the classifier learn slower. This can be recognized by comparing the batch sizes 8 and 64. Most interesting is that the batch size of 64 achieves the best mAP but decrease in the score after reaching the maximum. This drop in performance results from the overfitting to the

training data.

With a training set size for an individual teacher of 27438 samples in the "big 10" data, a batch size of 64 and 10.000 iterations of training every instance is shown to the classifier about 23 times. The training process is stopped after this point is reached, and the best performing model is taken for the ensemble. A sample size of 64 seems reasonable in combination with batch balancing. As the reduced ontology has 153 classes, a batch would contain at least half of the classes present in the ontology. This helps to not prefer one class over another because of the high number of samples.



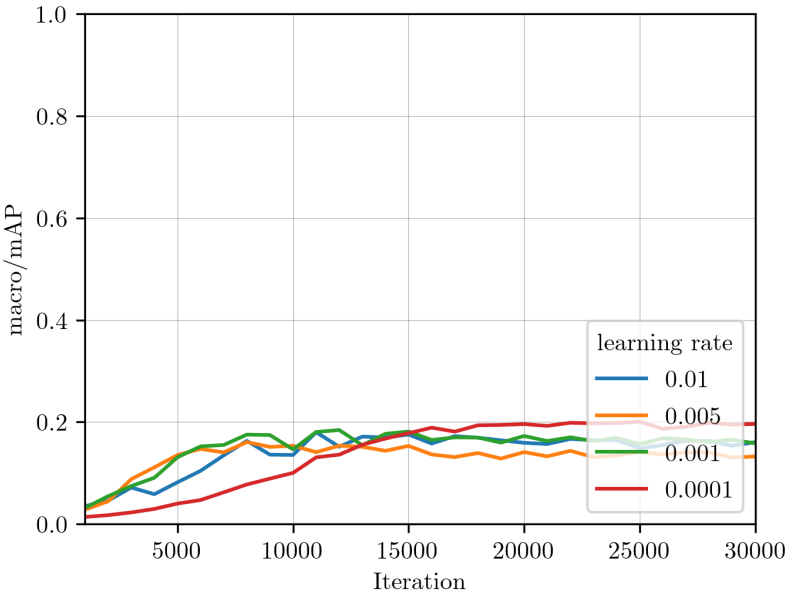
(a) mAP score



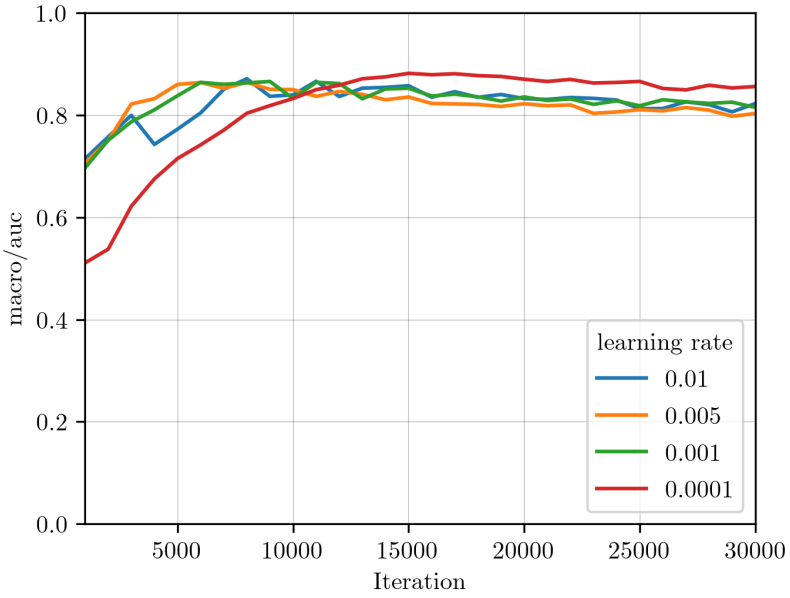
(b) AUC score

Figure 5.11: Performance of "big 10" data set for with and without adaptive weights in optimizer over iterations.

Not all classes are similar easy to learn. This brings up the idea to favor the classes with lousy performance in the loss calculation. Instead of calculating the loss mean for each sample in a batch, the samples are weighted according to their label assignments. This leads samples with not-so-good performing class labels to substantially impact the overall loss and, therefore, on the backpropagation step.
The weights applied to the class labels in the loss calculation are updated after a test step of the classifier according to the performance of the class on the test data. Such a feature can help for highly imbalanced samples. The comparison between regular loss calculation with BCE and the adaptive weighted version can be seen in the figure 5.11. It shows clearly that weight scaling in the loss function has no positive effect on the training result. This can be explained by the already large batch size and the other balancing features. With those, the loss depends on a high amount of samples and is more or less balanced among the classes. Additionally, with the high number of output classes, the weights on the classes tend to cancel out the effect of the weights. The loss weight scaling is not further considered in this work.



(a) mAP score



(b) AUC score

Figure 5.12: Performance of "big 10" data set for different learning rates over iterations.

The learning rate specifies how much the backpropagation step impacts the model parameters. A significant learning rate lets the model quickly adapt to the results of an iteration, but a too high learning rate can lead to a model flipping back and forth between two not locally optimal states. On the contrary, a too-small learning rate prevents a model from overcoming a local optimum as it is too cautious about changing parameters. In the figure 5.12, it can be easily observed that learning rates greater than 0.0001 let the model converge early, but the best performance is reached with the lowest evaluated learning rate. For a final model, a learning rate close to 10^{-4} is the best choice. For the curse of this thesis, partially more significant learning rates are used to profit from the faster training.

In this section, several features and parameters are presented that significantly impact the individual model and hence on the ensemble. The most promising features are "batch balancing" and the data augmentation technique "mixup". Any adaption directly in the loss function harms the training process. For training a model on the *Audio Set* a learning rate close to 10^{-4} in combination with a batch size of 64 or smaller should be used.

5.2 Teacher ensemble performance

The ensemble utilizes the knowledge of the individual teachers to create the aligned ensemble predictions. Hence the prediction of the ensemble is indirectly based on the entire data used for the student training. An aggregated prediction of the teacher ensemble should theoretically outperform the individual ones.
As the student is trained on predictions of the ensemble, it becomes an essential building block of the *PATE* framework. The better the ensemble can predict data, the better the student might perform. In the below evaluation and comparison, the knowledge learned by the ensemble is considered only from a performance perspective. All privacy-related characteristics will be discussed in the section 5.3.
Similar metrics can be used to evaluate the ensemble performance as for the teachers. Only the predictions do not originate from a individual model but the aggregation of the ensemble. In the *PATE* framework, non-private unlabelled data is used for prediction by the ensemble. To evaluate the ensemble performance, the same test set of the *Audio Set* is used as discussed in section 4.1. This allows us to compare the actual correct labels against the ensemble prediction. The ensemble predictions are bare boolean indicators if a label is present in the sample or not. This inherits from the aggregation function described in the section 4.3.4. Hence more granular metrics like mAP or ROC curves can not be used as they build on different threshold levels. The aggregated result only contains present or not present for privacy reasons. Anyway, the aggregation can be technically performed without applying the indicator function. If those results are used they are marked with "raw" in the below tables.
The first table 5.1 illustrates the performance of the aggregated ensemble votes compared to the baseline. A single teacher is trained on all available training data for the baseline with the same configuration as the teachers in the described ensembles. This baseline teacher predicted the same student samples as the ensemble. Hence the prediction quality can be compared between the baseline and the ensembles. For this comparison, zero noise is applied to the ensemble aggregation to avoid tampering of the results. The data is reported for the "big" model with 153 classes. A similar comparison for the "small" experiment set can be found in the appendix A.1.

#teacher	mAP (raw)	AUC (raw)	ensemble performance thresh.	ensemble performance P / R / F1	individual teacher (avg.) P / R / F1
baseline	0.268	0.923	adaptive	0.316 / 0.336 / 0.256	n/a
10	0.219	0.919	0.2	0.401 / 0.265 / 0.228	0.350 / 0.188 / 0.167
20	0.199	0.909	0.2	0.447 / 0.230 / 0.204	0.305 / 0.151 / 0.136
50	0.186	0.898	0.2	0.573 / 0.169 / 0.173	0.271 / 0.114 / 0.112

Table 5.1: The optimal performance of teacher ensemble compared to the average performance of the individual teachers on the "big" data set. No noise applied $\sigma = 0$.

The mAP and AUC scores can be reported independently from any specific threshold. A specific threshold must be chosen for the metrics precision, recall, and F1 score. For this purpose, the threshold with the best F1 score is selected. This ensures that the best balance between precision and recall is considered for the comparison. All metrics are macro-averaged to respect the different complexity of the classes in the result.
For macro-averaging, all metrics are calculated per class and then become averaged. This is important here as otherwise, the imbalance of the class labels would affect the results. The "ensemble performance" column reports the respective values for the ensemble while the "individual teacher (avg.)." averages the steals for all individual teachers of the ensemble. In the different rows of the table, the different ensemble sizes are shown as listed in the table 4.1. The larger the ensemble is, the smaller are the data slices the teachers are trained on.
With the increasing size of the ensemble, the prediction performance decreases. This results at least partially from the lower performance of the individual teachers. As observed in the right column of the table, the prediction capability of a teacher decreases with the amount of training data available per teacher. For the "big 10" experiment, the individual teacher is trained on about 27.000 samples while the "big 50" is only trained on about 7.000 samples. This lets the precision and recall drop similarly.
For the ensemble performance, this is different. The precision increases with the size of the ensemble while the recall drops. The capability to predict less frequent labels by the ensemble originates fully from the teachers. If they can not predict the presence of a label, the ensemble can not either.
More interesting is the behavior of the precision score of the ensemble. With the increase in teachers, the precision increases as well. The more teachers considered for the consensus, the more powerful it becomes. The ensemble outperforms the best baseline precision for all fixed ensemble thresholds th_e significantly. This positive effect can not compensate for the drop in the recall. The balanced F1 score decreases with the size of the ensemble.
Increasing the ensemble size has a positive impact on the performance if it can be ensured that the recall does not suffer too much. For the *Audio Set* unfortunately, the recall declines drastically if only subsets of the training data are used. If more training data is available and every teacher could be trained on a more significant subset, the ensemble might ultimately outperform the baseline.

#teachers	threshold	precision	recall	F1
baseline	adaptive	0.316	0.336	0.256
baseline	0.2	0.390	0.349	0.278
baseline	0.4	0.568	0.220	0.235
baseline	0.6	0.713	0.142	0.175
10	adaptive	0.224	0.311	0.187
10	0.2	0.401	0.265	0.228
10	0.4	0.551	0.170	0.193
10	0.6	0.683	0.108	0.143
20	adaptive	0.191	0.259	0.158
20	0.2	0.447	0.230	0.201
20	0.4	0.650	0.133	0.156
20	0.6	0.791	0.078	0.105
50	adaptive	0.180	0.186	0.136
50	0.2	0.573	0.169	0.173
50	0.4	0.781	0.090	0.118
50	0.6	0.880	0.053	0.078

Table 5.2: Performance of teacher ensemble for different thresholds and number of teachers on the "big" data set. No noise applied $\sigma = 0$.

In the table 5.2 different thresholds are compared for the ensemble sizes. A threshold of e.g., 0.2 requires 20% of the teachers of the ensemble to agree that a label should be assigned. Respectively a higher threshold requires more teachers to agree to consider a label present for a sample. This higher threshold should positively affect the precision but might prevent a label from being predicted as the consensus is too low. The discussion about thresholds is picked up again in the privacy evaluation section 5.3 as a strong consensus allows lower privacy budget consumption. Again zero noise is applied to evaluate pure ensemble performance. A similar table for the "small" experiment can be found in A.2.

All performance values depend on the ensemble threshold. This number describes the percentage of teachers that must agree to consider a label as present in a prediction. For example, the threshold of 0.4 corresponds to at least 40% of the teachers agreeing that this label should be assigned. The absolute amount of teachers then depends on the ensemble size. In table 5.2 the performance scores for different thresholds and ensemble sizes are listed.

The same effect as described for the comparison of the ensemble sizes can be seen here as well for all thresholds. As expected, for the actual threshold, the precision increases with the threshold while the recall drops. Anyway, the best F1 score of an ensemble 0.228 for the "big 10" experiment remains behind the best baseline configuration. The optimal performance regarding the F1 score is reached with the low threshold of 0.2, which corresponds to 20% of the teachers. This result is not the desired one as for the privacy analysis later a strong consensus among all teachers improves the privacy cost.

Even the adaptive threshold does not improve the ensemble predictions. The optimal threshold

will be discussed later with the missing metric of privacy exposure.

#teachers	threshold	confidence threshold 0.4		confidence threshold 0.6	
		#answ.	P / R / F1	#answ.	P / R / F1
10	adaptive	65%	0.416 / 0.314 / 0.269	48%	0.565 / 0.322 / 0.304
20	adaptive	62%	0.570 / 0.237 / 0.233	47%	0.660 / 0.251 / 0.256
50	adaptive	60%	0.739 / 0.184 / 0.211	44%	0.787 / 0.223 / 0.257
10	0.2	65%	0.441 / 0.370 / 0.283	48%	0.506 / 0.390 / 0.308
20	0.2	62%	0.497 / 0.304 / 0.252	47%	0.548 / 0.325 / 0.274
50	0.2	60%	0.633 / 0.247 / 0.243	44%	0.667 / 0.291 / 0.284
10	0.4	n/a		48%	0.602 / 0.322 / 0.309
20	0.4	n/a		47%	0.696 / 0.260 / 0.266
50	0.4	n/a		44%	0.799 / 0.232 / 0.266

Table 5.3: Performance of teacher ensemble for different confidence levels and several teachers on the "big" data set. No noise applied $\sigma = 0$.

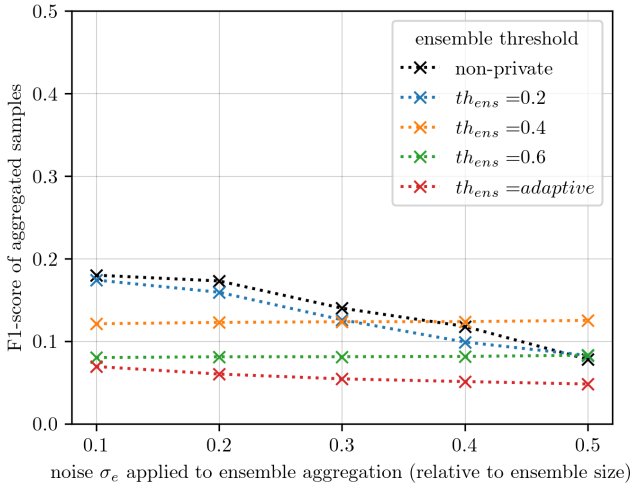
To improve privacy consumption, the confidence mechanism is discussed in the section 4.2.4. It allows to select and answer only the samples by the ensemble where a minimum agreement in the ensemble exists. Here, the aggregation mechanism is run twice to select the answered samples and then to predict those. This has a positive impact on the performance as complex samples are dropped and do not count into the evaluation metrics, but the percentage of samples that become predicted drops with the minimum agreement hurdle.
In the table 5.3, the respective precision, recall and F1 score metrics are reported for the two confidence threshold levels 0.4 and 0.6. This means that 40% or 60% of the teacher must agree on at least one label that the ensemble tries to predict. The new "#answ." column describes the percentage of predicted samples out of all available samples for the ensemble.
It can be seen that the overall performance improves with the confidence aggregator. This is expected as the aggregation mechanism filters the samples that allow a strong consensus among the teachers. For the confidence threshold of 0.4, slightly less than two-thirds of the samples allow such a consensus, and for the confidence threshold of 0.6, this applies to a little bit less than half of the samples.
Performance-wise the confidence aggregator allows the ensemble to outperform the baseline. The best performing ensemble configuration with a threshold of 0.2 and ten teachers will enable a performance increase for the F1 score of 24% for the confidence threshold of 0.4 and 35% for the confidence threshold of 0.6. Even more significant are the improvements for larger ensembles. For the "big 50" ensemble, the gains are 40% and 64% for the 0.4 and 0.6 confidence thresholds. While the performance increase, the percentage of answered queries drops with a high confidence threshold. As those samples might be missing, then later for the training of the student model, the optimal trade-off between those two things must be evaluated. This is later discussed in the section 5.4. The confidence aggregator has an impressive impact on the performance of the ensemble. Compared to the regular aggregation, it allows a more substantial average consensus and better predictions by actively selecting the samples worth answering.

As discussed earlier, the performance can heavily differ by class. It is interesting to evaluate the impact of the ensemble aggregation mechanism on individual classes. In the figure A.6 the five most and least frequent classes in the student training set are evaluated for their presence in the predictions. The blue bars are the absolute label occurrences in the student training set. As the student training data follows the same distribution of labels, this equals the label distribution in the training data of the teachers.
All other colored bars refer to actual predictions of the baseline or the teacher ensembles. For the baseline, it can be seen that there is no issue in predicting less frequent classes. The single model sufficiently learned to predict even less frequent classes without saying that those labels are correct. With the increasing ensemble size, it can be observed that the ensemble has problems predicting those less frequent classes at all.
In the sub-figure for the "big 50" experiment A.6, it can be observed that no predictions are made for the less frequent labels at all. This goes back to the training of the teachers themselves. With the reduced number of training samples, they do not learn to predict some classes. At least not in a way that 20%, 40%, or 60% of the teachers of the ensemble agree on such a label. This becomes problematic for the student if there are no positive examples for a class. It can not build up any knowledge to predict this class. The recall for this class of the student model is always zero.
Only for the adaptive threshold mechanism tends to predict some classes. This is because the adaptive mechanism favors any count of teacher votes assigned to a class if it is above the median aggregated votes for this sample. No fixed threshold of teacher votes is required.
On the contrary, the precision scores of the adaptive mechanism are worse than for the basic threshold aggregation. Hence the adaptive mechanism predicts those rare classes, but overall it creates more incorrect predictions. This can be beneficial for the student training as otherwise there are no samples for some of the classes. This is further evaluate in the student evaluation in section 5.4.
The teacher ensemble's performance depends on the ensemble's size and the prediction quality of the individual teachers in the ensemble. The larger the ensemble is, the performance drops. This results mainly from the falling recall due to training the teachers on smaller data sets.
If there is a label assignment made by the ensemble, it profits from a large ensemble. The more teachers it has, the higher is the precision. None of the ensembles can outperform the baseline. Only if the selective confidence aggregator is used the ensemble's performance increases significantly. But this goes at the expense of the samples that are predicted. Up to half of the samples are not predicted and hence unavailable for student training later. This has an extreme impact on classes that are hard to predict and are less often available in the label distribution. Such classes can not be learned by the student model if the ensemble can not predict them.

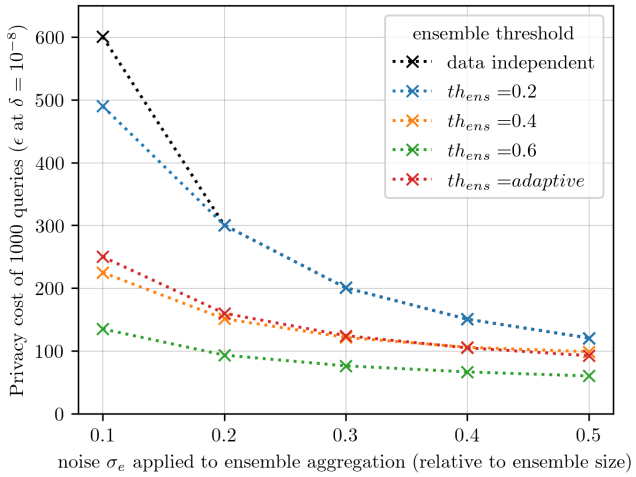
5.3 Privacy analysis of ensemble predictions

The privacy analysis of the teacher ensembles is the key contribution of the *PATE* framework. It allows quantifying the consumed privacy to train the student model. The student training is discussed later in section 5.4. The privacy analysis can be entirely based on the teacher ensemble without knowing the student model. For the student's training, a fixed number of training

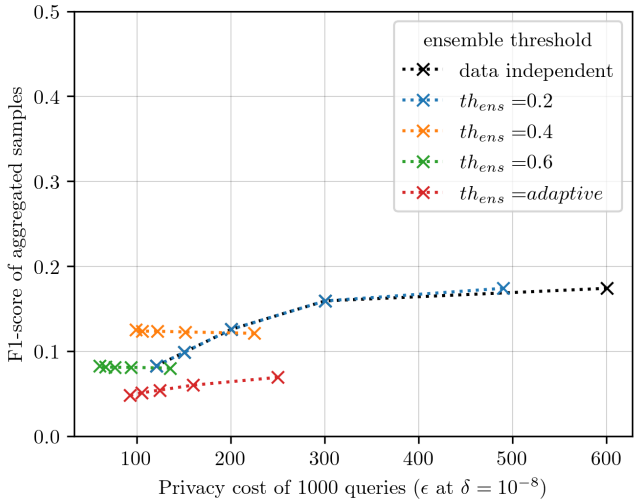
samples labeled by the ensemble are used. Due to this scoping, the consumed privacy budget for any student model trained on this fixed set of predictions is the same.
The consumed privacy budget can be compared against two baselines, on the one hand against a non-private mechanism and a data-independent one. As RDP does allow to calculate the spent privacy only for somehow private algorithms that have noise $\sigma > 0$, there is no measurable non-private baseline. Instead, the relative improvement over a nearly non-private baseline can be measured. For this the relative improvements between the different noise levels can be used. The non-private baseline allows comparing the improvement of the *PATE* mechanism over a fundamental tool.
The prediction performance must be balanced with the spent privacy budget for the privacy analysis. Increasing privacy always is at the expense of the prediction performance of the classifier and the other way around. The scores depend on the number of teachers, the applied noise, and the aggregation methodology. For the privacy analysis and discussion, mainly the "big 50" experiment is used. For comparison with other experiments, similar plots can be found in the appendix. It can be easily observed that the number of teachers has a considerable impact on privacy consumption. The original *PATE* work was analyzed with multiple hundred teachers. The *Audio Set* data is too small for many such teachers.
The figure 5.13 shows the key figures for the base aggregation method without a confidence threshold. The first two sub-figures present the F1 score and the averaged privacy cost for the first 1000 queries according to the noise applied to the ensemble aggregation.
The privacy cost for for the first 1000 samples is summed in the RDP domain and later transferred to corresponding (ϵ, δ)-DP. This corresponds to the overall consumed privacy for a student trained on only 1000 queries to the ensemble.
In the third figure, the F1 score and the privacy cost are shown relative to each other to simplify the two characteristics. The corresponding values for σ_e can be retrieved from the other two graphs. The different lines represent different thresholds of the ensemble that must be crossed to consider a label present. For example the $th_e = 0.2$ corresponds to 20% agreeing on a prediction.
None of the noisy ensemble aggregations outperform the non-private baseline without any noise. It can be observed that for the prediction performance, a low threshold with minimum noise results in the best F1 score. Increasing the noise for the low ensemble threshold $th_e = 0.2$ lets the F1 score drop rapidly. If the applied noise outweighs the threshold, the predictions become random. Hence the applied noise should not be larger than the threshold applied used for the ensemble aggregation. For thresholds, $th_e = 0.4$ and $th_e = 0.6$, the observed noise scales have nearly no impact on the prediction performance. For the adaptive threshold, the performance slightly drops as the gaps between positive and negative predictions become randomized.
The difference between the various ensemble thresholds th_e is more significant for the privacy cost. The higher the ensemble threshold is, the lower the privacy cost is. As discussed earlier, the consensus often is fragile for the *Audio Set* task. The distance between the threshold and those weak predictions are bigger with the larger threshold. Even if some correct predictions are missed, a stronger consensus that a label is not present allows stronger privacy guarantees.
For the smallest observed threshold $th_e = 0.2$, the privacy consumption is nearly equal to the data-independent baseline. Hence the effect of the data-dependent privacy analysis of the *PATE* mechanism does not outperform the basic privacy analysis.



(a) F1-performance



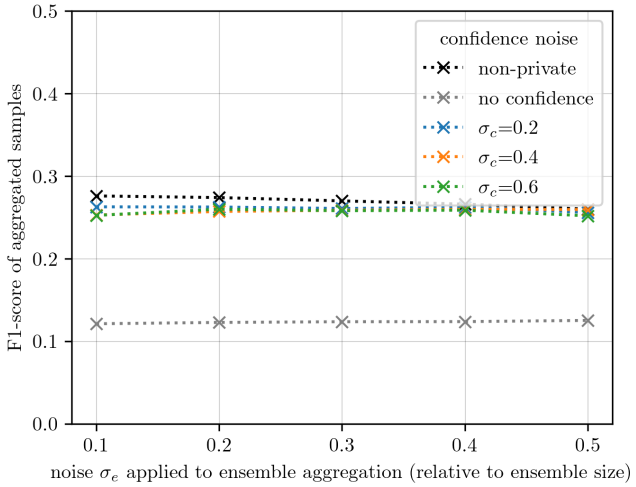
(b) privacy consumption



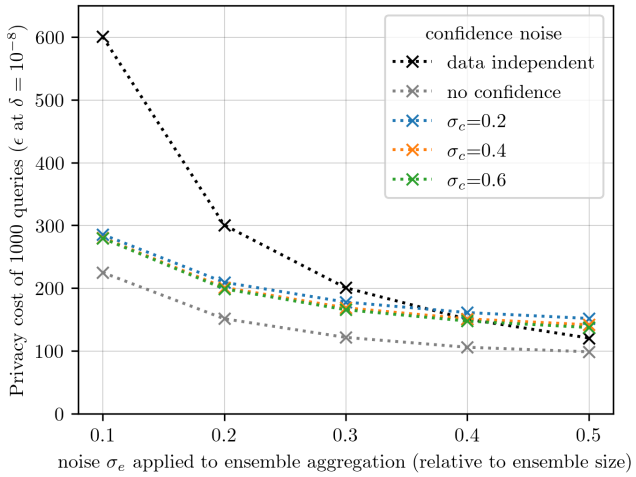
(c) combination

Figure 5.13: Ensemble performance for different noise scales and thresholds on the "big 50" experiment.

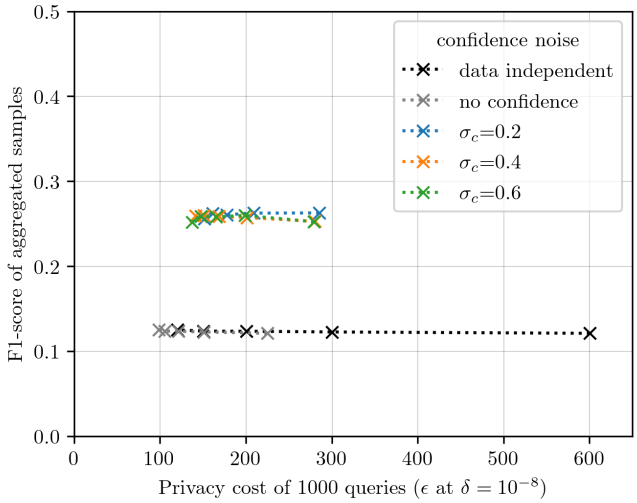
The decrease in privacy cost is most significant for the step from noise $\sigma_e = 0.1$ to $\sigma_e = 0.2$. The privacy is nearly cut in half. Further increasing the noise reduces the privacy cost less strong. It can be assumed that the real non-private privacy costs are way higher than the values for $\sigma_e = 0.1$.
The privacy cost of the adaptive aggregation and the $th_e = 0.4$ are close to each other. Only the maximum observed threshold $th_e = 0.6$ helps to reduce the privacy cost even further but requires 60% of the teachers to agree on a vote.
Bringing both metrics together shows how performance and privacy costs are balanced for the practical experiments. The optimal value would be in the top left corner with a maximum F1 score and close to zero privacy cost. For the presented values $th_e = 0.4$ and a relatively high noise for the teacher ensemble aggregation $\sigma_e = 0.5$, a considerably high prediction performance with a low privacy cost can be achieved. Higher performance can be achieved only with about three times the privacy cost. Optimizing the privacy cost is only possible by losing about a third of the prediction performance.
A simple approach to improving the predictions' quality is to separate the samples into easy and hard to predict. Samples with a strong consensus among the ensemble allow smaller privacy costs. The *PATE* mechanism describes this in the confidence aggregator mechanism. Here the samples first are passed through a separator that estimates if there can be a strong prediction for this sample. Only if this is true, the ensemble performs the actual prediction. As both activities use the predictions, both have to be considered for the privacy analysis.
In figure 5.14 the metrics of the confidence mechanism are shown for a fixed ensemble threshold $th_e = 0.4$ and an even more strict confidence threshold $th_c = 0.6$ on the "big 50" experiment. Other threshold configurations can be found in the appendix A.10 and A.11. The axes and plots correspond to the ones in the regular fixed threshold aggregator. The different colors no longer represent a teacher ensemble threshold but the noise applied to the confidence aggregator. Each set of those three graphics has a fixed threshold th_e for the ensemble and the confidence check th_c.
The performance, in general, is strongly impacted by the confidence mechanism. Compared to the aggregation without confidence check, the F1 score doubled. The confidence check is a practical approach to remove privacy-wise harmful or hard to predict samples. Without the confidence check, none of the configurations can outperform the non-private baseline that uses no noise. Interestingly, the performance is nearly stable for adding higher portions of noise to the confidence check.
From the privacy perspective, the confidence mechanism evaluations shows the downside of the additional check. The extra query performed on the data must be considered for the privacy analysis. All confidence mechanisms have a higher cost of privacy than those without the additional confidence mechanism. This inherits from the privacy cost that must be added for every sample due to the initial confidence check. The decreased privacy cost due to the advantage that only samples with expected strong confidence are predicted can not outweigh the additional privacy cost from the confidence check. As expected, the privacy cost behaves reciprocally to the amount of applied noise. The more noise is applied, the privacy cost decreases. But the difference between the privacy costs for doubled or tripled noise is minimal.
Both metrics combined again allow us to compare the privacy cost with the performance. Using



(a) F1-performance



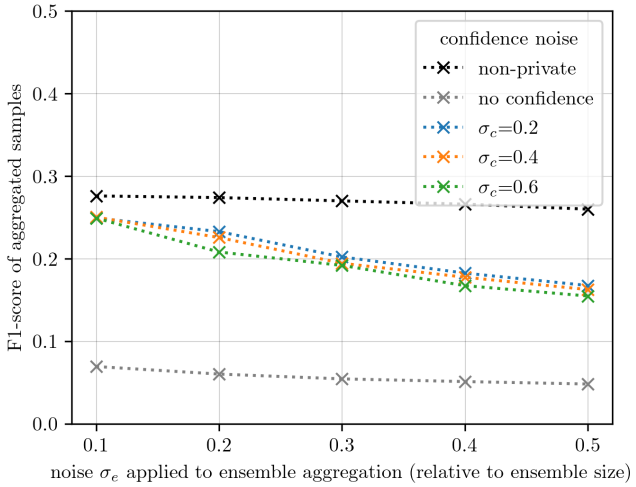
(b) privacy consumption



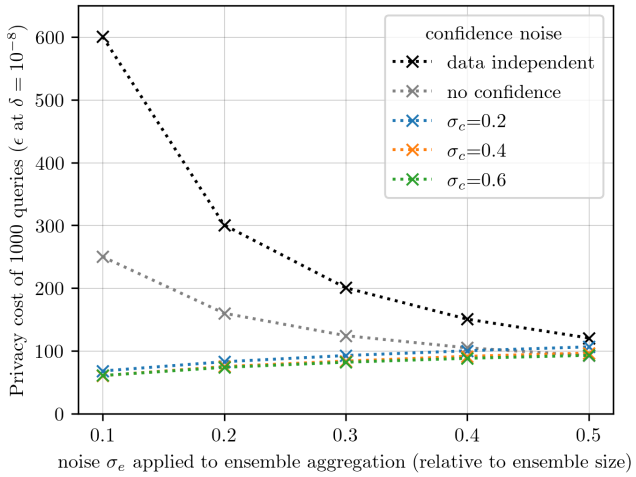
(c) combination

Figure 5.14: Ensemble performance for different confidence noise scales on the "big 50" experiment with $th_e = 0.4$ and $th_c = 0.6$.

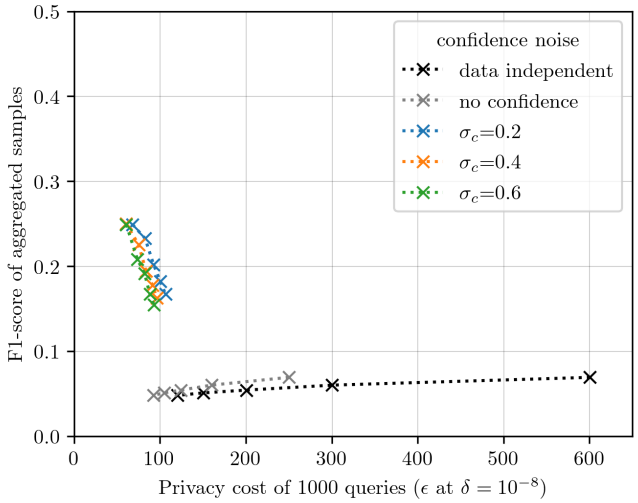
the confidence mechanism in this configuration results in a more than double F1 score with a worsened privacy cost according to the factor of 1.5. Overall the confidence mechanism allows significant stronger F1 scores at the expense of moderately increased privacy cost.



(a) F1-performance



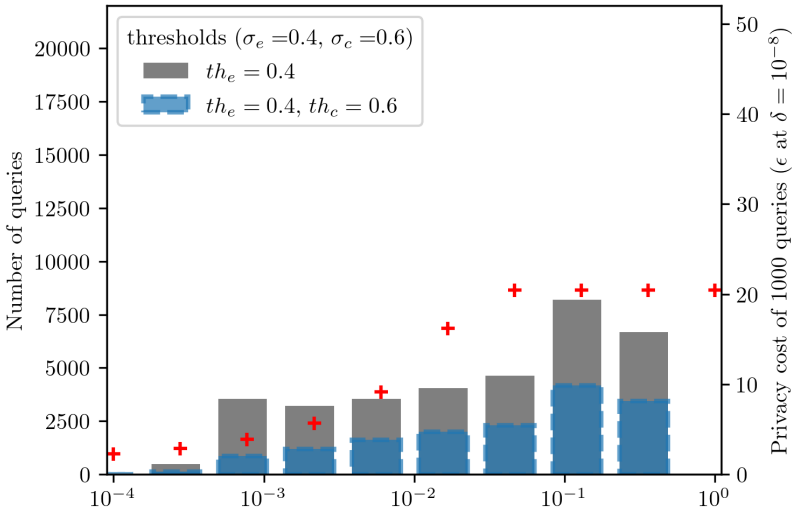
(b) privacy consumption



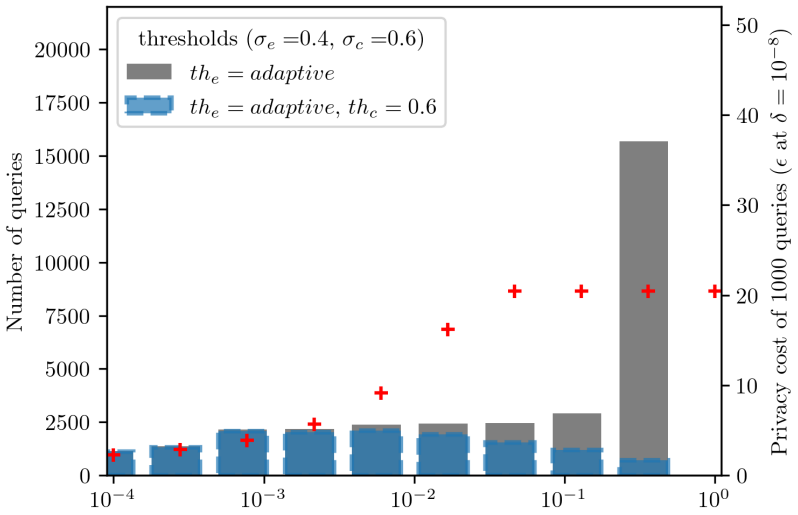
(c) combination

Figure 5.15: Ensemble performance for different confidence noise scales on the "big 50" experiment with adaptive ensemble threshold and $th_c = 0.6$.

The confident mechanism can demonstrate its full strength combined with the adaptive aggregation mechanism for the actual ensemble vote. In the figure 5.15 this configuration can be observed. The threshold confidence $th_c = 0.6$ is fixed while the threshold for the ensemble itself uses the described adaptive approach.
The best prediction performance and the lowest privacy cost this configuration can demonstrates with an ensemble noise $\sigma_e = 0.1$. This is quite surprising as, in all other scenarios, the privacy cost decreases with the amount of applied noise. It is the exact other way around for this configuration, but it can be explained.
The confidence check ensures at least one label with a strong consensus among the teachers. In the shown configuration, this is the case if 60% of the teachers agree on a vote. Such samples are then handed over to the actual prediction that uses the adaptive threshold mechanism. The adaptive part tries to find a threshold that maximizes the distances between the positive and negative predictions. For example, if the confident label aggregates votes from 70% of the teachers and the next highest label score is 10%, the adaptive threshold is 40%.
The privacy cost depends on the distance between the threshold and the percentage of teachers that agree to each label and the noise parameter. In this scenario, the described distance is perfectly optimized, making the noise parameter less critical. On the contrary, increasing noise shortens the distance and reduces the effectiveness of the adaptive mechanism.
The adaptive mechanism can demonstrate this strength only in combination with the confidence check. Without the confidence operation, the adaptive mechanism performs worse than the simple threshold mechanism, but if the samples are selected already, it predicts equally strong as the fixed threshold mechanisms.
In addition, it allows a far lower privacy cost. Even the additional privacy cost of the confidence check can be compensated, and the curve is below the experiment without the confidence mechanism. In combination with the confidence check, the adaptive mechanism provided a similar prediction performance as the fixed threshold check and the non-adaptive ensemble aggregation, as shown in the figure 5.14. But only half the privacy cost is required compared to a fixed mechanism. This proves the confidence mechanism as a solid addition to the current *PATE* framework.



(a) $th_e = 0.4$



(b) th_e =adaptive

Figure 5.16: Distribution of q on the "big 50" predictions with averaged combined privacy cost for the confidence mechanisms ($\sigma_e = 0.4$).

This behavior can be observed additionally in a different representation. The figure 5.16 shows the frequency of samples for a set of buckets for different values of q and the averaged privacy cost for samples in this bucket by the red markings. The buckets are shown on a logarithmic scale and are equally sized in the logarithmic space. The buckets range from 10^0 to 10^{-4}.
As smaller q are better than values close to 1, the x-axis is plotted inverse. The number of samples in the bucket is shown on the left axis, while on the right, the averaged privacy cost for 1000 queries to samples of this bucket is shown. The figure is fixed for a specific ensemble threshold and noise scales. For the illustration here the noise is fixed to $\sigma_e = 0.4$ and $\sigma_c = 0.6$. Further noise constellations can be found in the appendix A.12 and A.13.
It can be in all figures of this type observed that the confidence mechanism answers fewer samples than the setup without a confidence check. This is the desired behavior as unclear predictions harm the privacy cost. But an extreme reduction of answered samples might cause problems, as discussed in the section 5.2.

Additionally, a barrier for q that separates if the data-dependent analysis allows stronger guarantees over the data-independent privacy analysis. The curve modeled by the red markings flattens immediately between 10^{-2} and 10^{-1} for the applied parameters. This is the break-even of the *PATE* data-dependent privacy analysis. If the value of q is less than this threshold of q, the privacy cost profits from the data-dependent analysis.
The top figure shows the distribution of q with a fixed ensemble threshold $th_e = 0.4$, while the bottom figure utilizes the adaptive mechanism $th_e =$ adaptive. The shaded blue in front of the grey bars displays the number for the confidence mechanism with similar parameters. The grey bar in the background shows the basic algorithm without the confidence check.
The most apparent difference between the adaptive and fixed ensemble threshold is that the adaptive mechanism has a significant peak on the right side with a significantly high q. This explains the high privacy cost of the adaptive mechanism. Instead, the fixed ensemble threshold is distributed right-hand-sided but less extreme than the adaptive threshold. For the regular fixed ensemble threshold on top, the distribution of q is nearly preserved. Only the number of answered samples got reduced.
Observing the bars of the confidence mechanism changes this picture a bit. The distribution is here rather left-hand-sided for the adaptive ensemble threshold, and the samples answered in the max-q bucket are nearly not answered anymore. This left-sided distribution explains why the adaptive ensemble mechanism and confidence check can achieve low privacy costs.
The privacy costs of predictions of a teacher ensemble are a manifold topic. The predictions, noise scales, mechanisms, and especially the predictors must be well-calibrated. In general, the better the prediction capabilities of the ensemble, the lower the privacy costs are. With tools like the confidence check, improvements of the predictions can be achieved by actively selecting the predicted samples. But specific combinations of configurations and parameters lead to surprising results.
As the *Audio Set* is rather a complicated prediction task, the agreement of the ensemble is very low in most cases. For the interpretation of the privacy costs, it must be kept in mind that for a multi-label prediction task, the presence of every label is sensitive information. Hence the absolute values can not be compared to most other works that observe multi-class scenarios.
Further privacy must be balanced with utility. Applying too much noise to a mechanism to achieve maximum privacy destroys the model's prediction capability. A combination of the adaptive ensemble aggregator and the confidence check shows the optimal result of privacy cost and utility for the discussed mechanisms. This configuration delivers low privacy costs with a minimal prediction performance decrease over the non-private baseline. Compared to the maximum observed data-independent privacy cost, the privacy cost of this configuration is twelve times less. As discussed in the preface of this section, there is no objective non-private baseline possible with the used tools of RDP. It can be assumed that the actual privacy saving over the non-private baseline is even higher.
It could be shown that the *PATE* framework privacy analysis can be successfully adapted to a multi-label scenario. Even on the complex task of the *Audio Set*, this toolset allows formulating strong privacy guarantees.

5.4 Student training and performance

The training of a student model is the final step of the *PATE* framework. For the student, non-private samples labeled by the teacher ensemble are used to train a new model. This model can be of any type and with any already discussed feature. The consumed privacy budget derives only from the predicted samples of the ensemble required for training the student. Hence the training process shall use as few samples as possible for the student's training. The overall goal is to train a student model with strong prediction performance while consuming as little privacy budget as possible.

The following examples of student models are trained on the "big 50" experiments in the following figures. The best parameters evaluated for the training of an individual teacher were transferred to the training of the student models. For details, refer to section 5.1.3.

The plots compare the achieved performance described by the mAP over the consumed privacy in terms of (ϵ, δ)-DP with $\delta = 10^{-8}$. Each plot represents a configuration of a teacher ensemble and confidence threshold. The plotted markers stand for a specific number of training samples n used for the student training and noise configuration relative to the number of teachers in the ensemble. At the same time, σ_e is the noise scale for the actual aggregation, and σ_c is the noise scale for the confidence mechanism. Those configurations significantly impact the performance and privacy cost of training a student model.

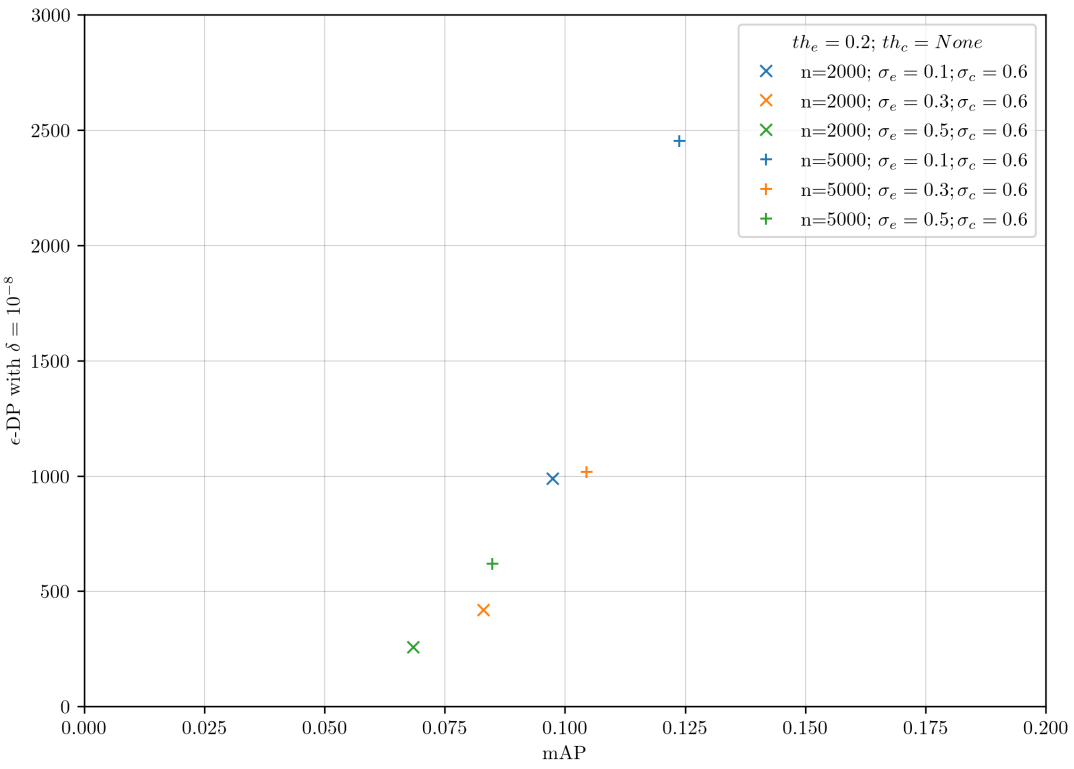


Figure 5.17: Comparison of different configurations of the student model for privacy cost and prediction performance with ensemble threshold $th_e = 0.2$ and no threshold mechanism.

The figure 5.17 illustrates the results of the ensemble configuration of $th_e = 0.2$ without any confidence check. This requires 20% of the teachers in the ensemble to agree on a vote. Privacy-wise, an individual teacher vote is hidden behind at least nine other teacher votes for the ensemble of 50 teachers. The prediction performance is only plotted up to a mAP of 0.2 as no values close to are retrieved in the evaluation.
The general observation from this plot is that the prediction performance can be increased at extend of additional privacy cost. The more samples are used for the training, and the less noise is applied to the aggregation, the better the newly trained student model performs. All achieved prediction performance values, under consideration of an appropriate privacy guarantee, are far below the state-of-the-art mAP-achievements for the *Audio Set* as a mAP 0f 0.314 [29].
The privacy cost to train those models is neither good nor bad. Considering that each sample has 153 boolean indicators, if a label is present or not, the privacy cost is relatively low for multiple thousand samples. On the contrary, if the individual teacher models would be able to agree to most of the predictions confidently, the privacy cost could be much less with the *PATE* framework privacy analysis.
Comparing the number of samples used to train the student model, it can be seen that privacy increases close to linear with the number of samples used. The privacy cost for 5000 samples corresponds to 2.5 times the privacy cost of the student trained on only 2000 samples. The prediction performance increases with additional samples as expected. Interestingly, the prediction performance profits significantly from a small noise scale for the teacher ensemble. The mAP score grows by about 20%, with 20% less noise injected than the ensemble size. While this behavior is expected in general, the effect is surprisingly significant. This can be explained if the ensemble calculates aggregated scores for the labels only minimum above the threshold.

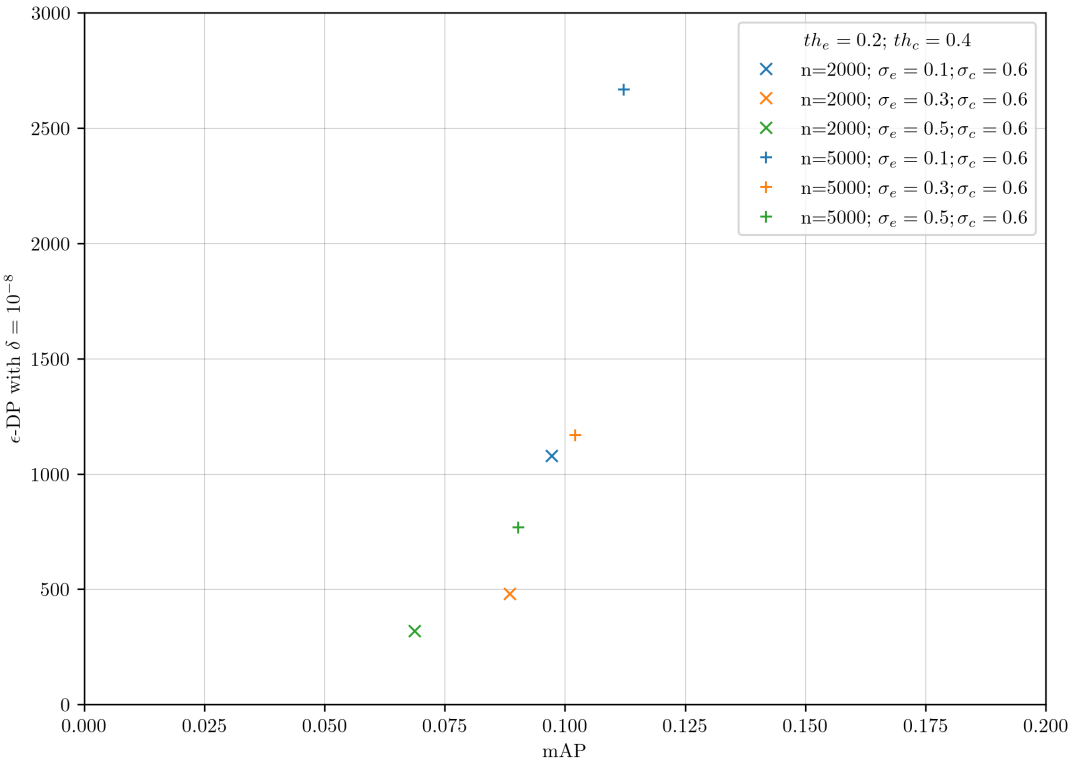


Figure 5.18: Comparison of different configurations of the student model for privacy cost and prediction performance with ensemble threshold $th_e = 0.2$ and fixed threshold mechanism $th_c = 0.4$.

The figure 5.18 shows similar configurations with added confidence checks. This additional operation sorts out samples that the ensemble has a weak consensus on, and hence they would have a high privacy cost. The downside of this mechanism is that while it prevents the high privacy costs for those samples, the check itself emits privacy cost by itself for each sample. For the fixed threshold mechanism, the question is if the advantage can compensate for the additional privacy consumption. This plot shows similar configurations as figures5.17 to evaluate this effect. In general, the data points show the same behavior as in 5.17.
The privacy costs are higher compared to the similar scenario without the threshold mechanism in addition. The additional privacy costs can not be compensated by removing weak-consensus samples. The effect promised by the *PATE* framework can not be realized due to a lack of performance in the individual teachers on the *Audio Set*. Even with the additional confidence check, the consensus of the teacher ensemble is not strong enough to minimize the privacy accordingly. A slight improvement can be seen for the compared configurations in the performance metric mAP. The additional confidence check reduces samples from the training set that are hard to predict in general. Removing these samples seems to improve the student training. This effect on the mAP score is less than 10% in all cases. In combination with the additional privacy consumed on the experiment data, it must be evaluated what metric should be optimized. This observation is shared with the results of sections 5.2 and 5.3. A slight advantage for the prediction performance goes at the cost of the privacy. The effect of *PATE* that the additional privacy is compensated can only be achieved with more robust individual teacher models.

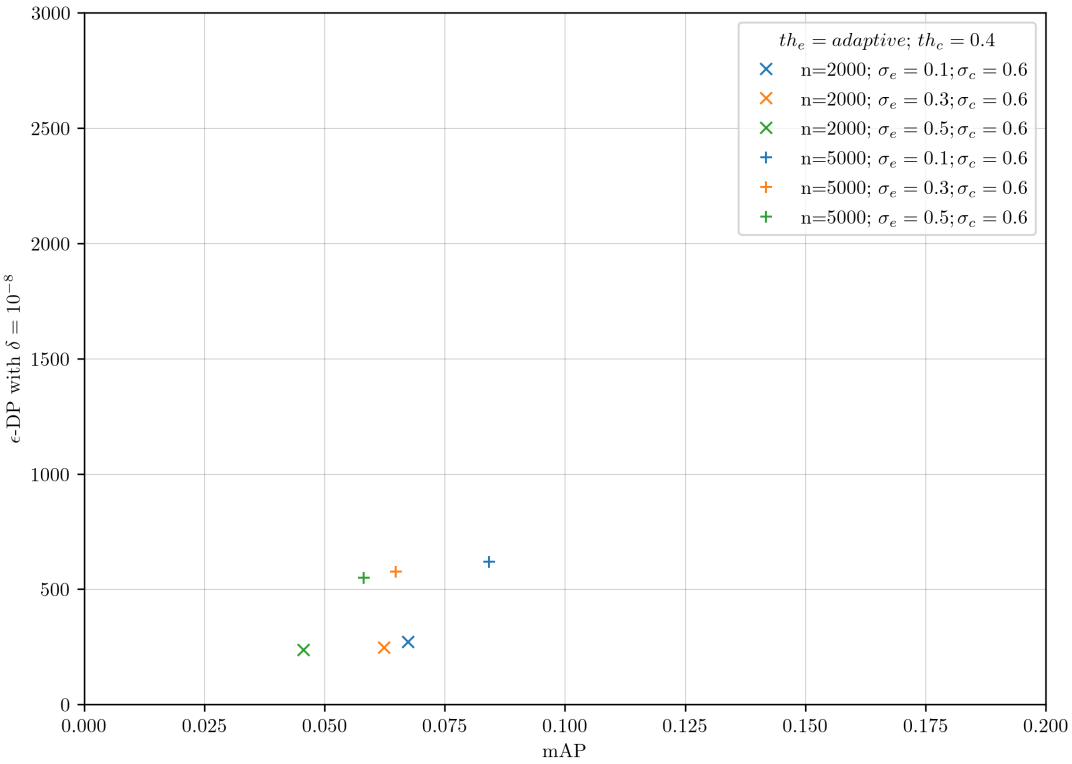


Figure 5.19: Comparison of different configurations of the student model for privacy cost and prediction performance with adaptive ensemble threshold $th_e = 0.2$ and fixed threshold mechanism $th_c = 0.4$.

The figure 5.19 uses the adaptive teacher ensemble aggregation instead of a fixed threshold. It must be seen in comparison to the two previous figures, 5.17 without any confidence check and 5.18 with a fixed confidence check.

This plot tries to answer if the optimal configuration for the teacher ensemble of sections 5.2 performs best for the student as well.

For privacy cost, this conflicts with the results. Choosing the adaptive threshold mechanism allows for much stronger privacy guarantees than similar aggregation configurations. Unfortunately, this is not true for the prediction performance mAP. The mAP score is less than for the two other configurations plotted in figures 5.17 and 5.18. While the adaptive mechanism reduces the privacy cost, it harms the prediction performance.

$\mathbf{th_e}$	$\mathbf{th_c}$	**data-dep.** (ϵ, δ)**-DP**	**data-indep.** (ϵ, δ)**-DP**	**q average**	**2000** **mAP**	**macro/F1**	**micro/F1**
0.2	None	418.4	418.4	0.747	0.083	0.082	0.189
0.2	0.4	479.5	522.1	0.759	0.088	0.087	0.191
0.2	0.6	481.1	568.2	0.770	0.078	0.076	0.174
0.4	None	268.4	418.4	0.156	0.072	0.062	0.178
0.4	0.4	371.4	524.4	0.193	0.074	0.066	0.183
0.4	0.6	347.4	566.6	0.164	0.071	0.063	0.180
adaptive	None	269.9	418.4	0.396	0.029	0.030	0.096
adaptive	0.4	248.5	523.5	0.132	0.062	0.055	0.161
adaptive	0.6	180.2	567.6	0.034	0.066	0.059	0.171
					5000		
0.2	None	1018.4	1018.4	0.748	0.104	0.097	0.214
0.2	0.4	1167.6	1277.7	0.760	0.102	0.103	0.208
0.2	0.6	1169.6	1384.2	0.768	0.098	0.093	0.182
0.4	None	631.2	1018.4	0.148	0.085	0.074	0.201
0.4	0.4	880.4	1277.8	0.189	0.096	0.084	0.206
0.4	0.6	839.1	1382.5	0.166	0.090	0.082	0.199
adaptive	None	632.3	1018.4	0.386	0.026	0.030	0.071
adaptive	0.4	577.1	1276.9	0.134	0.065	0.065	0.159
adaptive	0.6	413.5	1381.2	0.038	0.075	0.072	0.184

Table 5.4: Student model privacy and performance indicator for various configurations; with fixed noise levels for the ensemble $\sigma_e = 0.3$ and the threshold mechanism $\sigma_c = 0.6$; privacy values are reported with $\delta = 10^{-8}$.

In addition to the explanations for this behavior in sections 5.2 and 5.3 the table with the summarized results of the trained student model configuration help to understand the results. It shows the privacy and performance metrics for different ensemble and confidence thresholds for the evaluated number of samples used for the training.

The data-dependent privacy corresponds to the (ϵ, δ)-DP privacy cost of the trained student evaluated with the adapted *PATE* framework of section 4.2. In contrast to this, an essential privacy evaluation without using the specifics of the data is available for comparison in the data-independent privacy column. It can be observed that no difference exists for the configuration with a 20% ensemble threshold and no confidence check. This means the benefit of the adapted *PATE* multi-label privacy analysis does not exist. With a higher noise scale, the effect of the data-dependent privacy analysis can be observed. Hence the multi-label *Audio Set* task profits from the new *PATE* based privacy analysis.

The q describes the averaged probability over all samples that the most likely result will not be outputted due to noise as explained in section 4.2. With the usage of the confidence checks, the average of q drops significantly, as already discussed in section 5.2. The average of q is not close to the break-even value of q, which would significantly improve the privacy cost of the student model.

An approach to explain why the prediction performance is not better is already discussed in

the section 5.2. The big problem is the performance of individual classes. The ensemble has difficulties predicting classes the individual teachers struggle with already. This can be observed in the table by comparing the micro and macro averaged F1 scores.
As described in the section 2.1.3, the macro evaluates each class label individually and averages afterward, while the micro version evaluates all classes. The vast difference between both values illustrates the performance imbalance for different classes. This is the main reason for the student not to perform better.

5.5 Discussion

Evaluating the student model is the last step in building a classifier based on the adapted *PATE* framework. It can be summarized that the adapted *PATE* framework for multi-label classification could be successfully applied to the *Audio Set*. The results look promising, and the indications that the approach work as in the original *PATE* papers [67] and [68] could be demonstrated. The more difficult task of the *Audio Set* makes it harder to achieve good performance and privacy objectives. Substantial privacy improvement can be expected if future models and features significantly improve the individual classifiers based on the audio data. All tried privacy-preserving approaches lead to a significantly worsened performance of the classifier. The best-achieved scores with a considerable good privacy guarantee have about mAP$= 0.1$, which is compared to the initial baseline of 0.314 [29] and the current state-of-the-art of 0.439 [47] far lower.
This is the first work that transfers the *PATE* framework to a large-scale multi-label scenario. While the approach is agnostic for machine learning architectures and the underlying data, the aggregation mechanisms and privacy analysis were defined only for multi-class classification. With the presented adaptations, data curators can use the idea of private aggregation of teacher ensembles for more general classification tasks.
On the other hand, the application to the *Audio Set* demonstrated the weakness of the *PATE* approach. If the teachers in the ensemble do not agree strongly, only weak privacy guarantees can be given. In this work, the main limitation was that the training data must be split into small sets for each teacher.
Other works might be able to give stronger privacy guarantees due to these limitations. The *PATE* framework can provide strong privacy if there are at least 50 well-trained teachers. It might be challenging to find more than 50 data owners that want to cooperate in a real-life scenario. They must agree on formats and a standard class structure.
Depending on the domain, publicly available non-private training data for the student model can become a challenge. The group around [94] describe an alternative to *PATE* that utilizes a kNN-approach to pull the training information for the student mode from a shared data store and analyze the privacy cost based on the kNN step. They apply for their work only to multi-class and other data set as MNIST [16] and CIFAR-10 [49] and hence can not be compared to this work. Gitiaux et al. used the *Audio Set* to obfuscate another private speech data set. This does hold for an analogy to the work here.
Green and Plumbley aimed to train a privacy-preserving audio classifier on the much smaller FSD50K data set [26] using federated models. Even if they have neither used the *PATE* approach

for the aggregation of federated predictions of samples nor they provided a prof privacy framework their work is close to this thesis. Instead of a teacher ensemble the used a different approach to update the decentralized models„ by a centralized entity and analyzing the privacy based on these messages to the decentralized models as described in [89]. As they have not worked with the *Audio Set* or provided any privacy guarantees, no comparison of the performance or privacy is possible. Interestingly they see a drop in performance by increasing the number of federated models, which coincides with the thesis results.
Many other implementations and approaches of federated and private learning from decentralized data exist. Unfortunately, none of them is applied to the large-scale audio classification task of the *Audio Set*. *PATE* is the only framework that keeps the private data and models entirely with the data owners. Only samples that are non-private are sent over for labeling to the ensemble. The only private information is transferred together with the labeling of these samples from the knowledge of the teacher ensemble to the student model during training. This reduces the available attack surface significantly and makes a lot of attack scenarios impossible as nearly no private information is sent along communication channels. There is no similar approach built in the idea of preserving privacy classification in its high-level framework and design. *PATE* allows not only tight data-dependent privacy analysis but generates privacy by its structure already.
If it is possible to overcome the problem of the required number of teachers and the availability of non-private training data in a real-life scenario, *PATE* is a much more robust framework than most other proposed approaches. The possible privacy guarantees depend only on how good the individual predictors can become.

6 Conclusion and Outlook

Our economy has become more and more data-centric. Data is one of the key drivers for development, research, and growth. There are no limitations with the technical capabilities to record and gather everything everywhere combined with the computing power to handle this amount of data. However, the big question around data is - who owns it. Data can become quickly personal if it is a name, a pattern, a photo, or a conversation. There is always an individual that does not want to share this information with everyone.

Protecting the privacy of individuals while using their data in machine learning is the discipline of privacy-preserving machine learning. The *PATE* framework of Papernot et al. is a technique to learn from private with a bounded privacy guarantee [67]. This thesis extends the work of *PATE* to support multi-label classification. The existing aggregation methods are adapted, and a new adaptive aggregator is introduced combine the votes of the teachers. The GNThreshold Adaptive aggregator picks a threshold that minimizes the probability that the ensemble prediction is flipped by the application of noise.

The main achievement of the original work, the RDP-based *PATE* privacy analysis, is transferred to support multi-label classification. The quantification of the privacy cost is based on a data-dependent analysis for the query predictions by an ensemble. The adapted framework can now be used for sparsely labeled multi-label data.

To prove the utility of the new approach, an audio classifier under consideration of the adapted *PATE* framework is implemented. For this task, the *Audio Set* library of two million sound snippets is used. The overall ontology of more than 500 classes was reduced to a subset of 153 classes belonging to the topic of city and environmental sounds. A subset of about one million samples is crawled and used to train the ensemble of teachers and the student model. The classification models are evaluated for individual teachers, combined as the aggregated ensemble predictions, and after the knowledge transfer to the student.

For the individual classifiers, various features are evaluated. The main challenge for the ensemble's training is to train an audio classifier for 153 classes on only a few thousand sound snippets. The most effective techniques are the data sampling strategy *batch balancing* and the data augmentation technique *mixup*. Further hyperparameters are analyzed and discussed to find an optimal configuration. With all optimizations, the performance of a model trained on the whole training data consistently outperforms ($mAP = 0.44$) the aggregated non-private ensemble ($mAP = 0.27$) for the aggregated votes of 50 teachers. This significant decrease in the prediction capability deduces that it is tough to learn from just a few thousand samples of sparsely labeled audio clips. Only the confidence checks allow a better prediction performance by actively selecting the samples that the ensemble can predict with sufficient confidence.

Various experiment configurations of up to 50 teachers are trained to analyze the ensemble performance. Based on those models, the different aggregation methods are analyzed for prediction capabilities and privacy costs. The poor prediction strength of the teacher models is a challenge for the ensemble and the resulting privacy cost. The lower the consensus in the ensemble is, the higher the privacy costs are. This does not fit well with the observation that ensembles with less than 50 teachers generally have worse privacy guarantees.

The most promising setup is the newly introduced GNThreshold Adaptive aggregator for the actual ensemble prediction combined with a relatively strict confidence check with a threshold of 60% of the ensemble. This configuration allows a privacy consumption twelve times less than the corresponding data-independent privacy cost while impacting the prediction capability only minimal with $mAP = 0.25$. Privacy and utility are always a trade-off. Hence a privacy-preserving model with a prediction capability close to the non-private baseline will be hard to build. If the private baseline is only compared to the best possible non-private aggregated result for 50 teachers, the performance in the mAP metric only decreases by 7%. However, compared to a single model trained on the whole *Audio Set* without splitting and aggregation, it is 43% worse.

Different student models got trained on the experiment sets. The performance metrics for the student model are in any configuration far worse than the state-of-art baseline. This results from the lack of prediction performance by the teacher ensemble and the significant differences between the performance for the classes.

It can be clearly shown that the data-dependent privacy analysis with the adapted *PATE* framework for multi-label classification allows much stronger privacy guarantees. Primarily the adaptive ensemble threshold results in very low privacy costs considering the information included in the private training data.

The adaptions can successfully extend the *PATE* framework to a multi-label scenario. Both key components, the aggregation of ensemble votes and the privacy analysis were extended to support multi-label classification. Building an audio classifier for the city and environmental sounds based on the *Audio Set* is challenging even without considering privacy. In order to give an acceptable privacy guarantee, the classifier's performance suffers significantly. While the concept is working well, future work must still improve the individual classifiers to increase the utility while preserving privacy. The strength of the data-dependent privacy analysis kicks in if the ensemble has a strong consensus. If the ensemble's performance can be further improved, stronger privacy guarantees are possible, and the prediction performance improves.

The key to improve the city and environmental sound classifier that respects private information in the samples is to enhance the individual teacher models. If it is possible to train even more than 50 teachers on small audio collections or increase the available audio data much tighter privacy guarantees can be achieved. Another big problem of the *Audio Set* and of the here built audio classifier are poor-performing classes. With progress on those individual points, the *PATE* framework is a powerful tool for an privacy-preserving audio classifier for the city and environmental sound.

Bibliography

[1] Martin Abadi et al. “Deep learning with differential privacy”. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM. 2016, pp. 308–318.

[2] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. `https://www.tensorflow.org/`. 2015.

[3] Gergely Acs et al. “Differentially private mixture of generative neural networks”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.6 (2018), pp. 1109–1121.

[4] Charu C. Aggarwal. “On k-anonymity and the curse of dimensionality”. In: *VLDB*. Vol. 5. 2005, pp. 901–909.

[5] Raef Bassily, Adam Smith, and Abhradeep Thakurta. “Private empirical risk minimization: Efficient algorithms and tight error bounds”. In: *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE. 2014, pp. 464–473.

[6] Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. “Plausible deniability for privacy-preserving data synthesis”. In: *Proc. VLDB Endow.* 10.5 (2017), pp. 481–492.

[7] Dan Bogdanov et al. “Privacy-preserving statistical data analysis on federated databases”. In: *Annual Privacy Forum*. Springer. 2014, pp. 30–55.

[8] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24.2 (1996), pp. 123–140.

[9] Forrest Briggs et al. “Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach”. In: *The Journal of the Acoustical Society of America* 131.6 (2012), pp. 4640–4650.

[10] Mark Bun and Thomas Steinke. “Concentrated differential privacy: Simplifications, extensions, and lower bounds”. In: *Theory of Cryptography Conference*. Springer. 2016, pp. 635–658.

[11] Emre Cakir et al. “Multi-label vs. combined single-label sound event detection with deep neural networks”. In: *2015 23rd European signal processing conference (EUSIPCO)*. IEEE. 2015, pp. 2551–2555.

[12] Mark Cartwright et al. “SONYC Urban Sound Tagging (SONYC-UST): A multilabel dataset from an urban acoustic sensor network”. In: (2019).

[13] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. “Differentially private empirical risk minimization.” In: *Journal of Machine Learning Research* 12 (2011).

[14] Kamalika Chaudhuri, Anand D. Sarwate, and Kaushik Sinha. “A Near-Optimal Algorithm for Differentially-Private Principal Components.” In: *Journal of Machine Learning Research* 14 (2013).

[15] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 248–255.

[16] Li Deng. “The mnist database of handwritten digit images for machine learning research”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[17] Dorothy E. Denning and Peter J. Denning. “The tracker: A threat to statistical database security”. In: *ACM Transactions on Database Systems (TODS)* 4.1 (1979), pp. 76–96.

[18] *Detection and Classification of Acoustic Scenes and Events*. `http://dcase.community/`. visited on 06.07.2021.

[19] P. Dhanalakshmi, S. Palanivel, and Vennila Ramalingam. “Classification of audio signals using SVM and RBFNN”. In: *Expert systems with applications* 36.3 (2009), pp. 6069–6075.

[20] Thomas G. Dietterich. “Ensemble methods in machine learning”. In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[21] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. “Privacy aware learning”. In: *Journal of the ACM (JACM)* 61.6 (2014), pp. 1–57.

[22] 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.

[23] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. “Boosting and differential privacy”. In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE. 2010, pp. 51–60.

[24] Cynthia Dwork et al. “Calibrating noise to sensitivity in private data analysis”. In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

[25] Nick Ficano. *pytube3 - A lightweight, dependency-free Python 3 library (and command-line utility) for downloading YouTube Videos*. `https://github.com/get-pytube/pytube3`. visited on 05.01.2021.

[26] Eduardo Fonseca et al. “FSD50K: an Open Dataset of Human-Labeled Sound Events”. In: *CoRR* abs/2010.00475 (2020).

[27] Logan Ford et al. “A Deep Residual Network for Large-Scale Acoustic Scene Analysis”. In: *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*. ISCA, 2019, pp. 2568–2572.

[28] 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*. ACM. 2015, pp. 1322–1333.

[29] Jort F. Gemmeke et al. “Audio set: An ontology and human-labeled dataset for audio events”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 776–780.

[30] *General Data Protection Regulation (GDPR)*. `https://gdpr.eu/tag/gdpr/`. visited on 06.08.2021. European Commission.

[31] Ran Gilad-Bachrach et al. “Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy”. In: *International Conference on Machine Learning*. PMLR. 2016, pp. 201–210.

[32] Yuan Gong, Yu-An Chung, and James Glass. "PSLA: Improving audio event classification with pretraining, sampling, labeling, and aggregation". In: *CoRR* abs/2102.01243 (2021).

[33] Guodong Guo and Stan Z. Li. "Content-based audio classification and retrieval by support vector machines". In: *IEEE transactions on Neural Networks* 14.1 (2003), pp. 209–215.

[34] Paul Gustafson, C. Srinivasan, and Larry Wasserman. "Local sensitivity analysis". In: *Bayesian statistics* 5 (1996), pp. 197–210.

[35] Jihun Hamm, Yingjun Cao, and Mikhail Belkin. "Learning privately from multiparty data". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 555–563.

[36] Jamie Hayes et al. "Logan: Membership inference attacks against generative models". In: *Proceedings on Privacy Enhancing Technologies (PoPETs)*. Vol. 2019. 1. De Gruyter. 2019, pp. 133–152.

[37] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015).

[38] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE. 2016, pp. 770–778.

[39] Shawn Hershey et al. "CNN architectures for large-scale audio classification". In: *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2017, pp. 131–135.

[40] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. "Cryptodl: Deep neural networks over encrypted data". In: *CoRR* (2017).

[41] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. "PATE-GAN: Generating synthetic data with differential privacy guarantees". In: *International conference on learning representations*. 2018.

[42] Georgios A. Kaissis et al. "Secure, privacy-preserving and federated machine learning in medical imaging". In: *Nature Machine Intelligence* 2.6 (2020), pp. 305–311.

[43] Taejun Kim, Jongpil Lee, and Juhan Nam. "Comparison and analysis of samplecnn architectures for audio classification". In: *IEEE Journal of Selected Topics in Signal Processing* 13.2 (2019), pp. 285–297.

[44] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[45] Qiuqiang Kong et al. "Audio set classification with attention model: A probabilistic perspective". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 316–320.

[46] Qiuqiang Kong et al. "Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems". In: *CoRR* abs/1904.03476 (2019).

[47] Qiuqiang Kong et al. "Panns: Large-scale pretrained audio neural networks for audio pattern recognition". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 2880–2894.

[48] Qiuqiang Kong et al. “Weakly labelled audioset tagging with attention neural networks”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.11 (2019), pp. 1791–1802.

[49] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

[50] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. “t-closeness: Privacy beyond k-anonymity and l-diversity”. In: *2007 IEEE 23rd International Conference on Data Engineering*. IEEE. 2007, pp. 106–115.

[51] Tian Li et al. “Federated learning: Challenges, methods, and future directions”. In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60.

[52] Lie Lu, Hao Jiang, and HongJiang Zhang. “A robust audio classification and segmentation method”. In: *Proceedings of the ninth ACM international conference on Multimedia*. 2001, pp. 203–211.

[53] Ashwin Machanavajjhala et al. “L-Diversity: Privacy beyond k-Anonymity”. In: *ACM Trans. Knowl. Discov. Data* 1.1 (2007), 3–es. ISSN: 1556-4681.

[54] Sébastien Marcel and Yann Rodriguez. “Torchvision the Machine-Vision Package of Torch”. In: *Proceedings of the 18th ACM International Conference on Multimedia*. MM ’10. Firenze, Italy: Association for Computing Machinery, 2010, pp. 1485–1488.

[55] Ueli Maurer. “Secure multi-party computation made simple”. In: *Discrete Applied Mathematics* 154.2 (2006), pp. 370–381.

[56] Brian McFee et al. “librosa: Audio and music signal analysis in python”. In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015.

[57] Martin McKinney and Jeroen Breebaart. “Features for audio and music classification”. In: (2003).

[58] Brendan McMahan et al. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1273–1282.

[59] Ilya Mironov. “Rényi differential privacy”. In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE. 2017, pp. 263–275.

[60] Payman Mohassel and Yupeng Zhang. “Secureml: A system for scalable privacy-preserving machine learning”. In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 19–38.

[61] Arvind Narayanan and Vitaly Shmatikov. “Robust de-anonymization of large sparse datasets”. In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE. 2008, pp. 111–125.

[62] Milad Nasr, Reza Shokri, and Amir Houmansadr. “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning”. In: *2019 IEEE symposium on security and privacy (SP)*. IEEE. 2019, pp. 739–753.

[63] Yuval Netzer et al. “Reading Digits in Natural Images with Unsupervised Feature Learning”. In: (2011).

[64] Christian Olms et al. “Architektur einer adaptiven Plattform für unternehmensübergreifende datenbasierte Dienste mit dem International Data Spaces”. In: *Logistics Journal: Proceedings* 2020.12 (2020).

[65] Sergio Oramas et al. “Multi-label music genre classification from audio, text, and images using deep features”. In: *arXiv preprint arXiv:1707.04916* (2017).

[66] Nicolas Papernot and Ian Goodfellow. *Privacy and machine learning: two unexpected allies?* `http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html`. visited on 01.04.2021. 2018.

[67] Nicolas Papernot et al. “Scalable private learning with pate”. In: (2018).

[68] Nicolas Papernot et al. “Semi-supervised knowledge transfer for deep learning from private training data”. In: (2017).

[69] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[70] Manas A. Pathak, Shantanu Rane, and Bhiksha Raj. “Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers”. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2010, pp. 1876–1884.

[71] Huy Phan et al. “Unifying isolated and overlapping audio event detection with multi-label multi-task convolutional recurrent neural networks”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 51–55.

[72] Karol J. Piczak. “ESC: Dataset for environmental sound classification”. In: *Proceedings of the 23rd ACM international conference on Multimedia*. ACM. 2015, pp. 1015–1018.

[73] Mohammad Al-Rubaie and J. Morris Chang. “Privacy-preserving machine learning: Threats and solutions”. In: *IEEE Security & Privacy* 17.2 (2019), pp. 49–58.

[74] Theo Ryffel et al. “A generic framework for privacy preserving deep learning”. In: *CoRR* abs/1811.04017 (2018).

[75] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. “A dataset and taxonomy for urban sound research”. In: *Proceedings of the 22nd ACM international conference on Multimedia*. ACM. 2014, pp. 1041–1044.

[76] Pierangela Samarati and Latanya Sweeney. “Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression”. In: (1998).

[77] Reza Shokri and Vitaly Shmatikov. “Privacy-preserving deep learning”. In: *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM. 2015, pp. 1310–1321.

[78] Reza Shokri et al. “Membership inference attacks against machine learning models”. In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 3–18.

[79] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: (2015).

[80] 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*. IEEE. 2013, pp. 245–248.

[81] Mohammad S. Sorower. “A literature survey on algorithms for multi-label learning”. In: *Oregon State University, Corvallis* 18 (2010), pp. 1–25.

[82] Google’s Sound and Video Understanding team. *Audio Set - A large-scale dataset of manually annotated audio events*. `https://research.google.com/audioset/`. visited on 01.02.2021.

[83] Latanya Sweeney. “k-anonymity: A model for protecting privacy”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.

[84] Latanya Sweeney. “Weaving technology and policy together to maintain confidentiality”. In: *The Journal of Law, Medicine & Ethics* 25.2-3 (1997), pp. 98–110.

[85] Aleksei Triastcyn and Boi Faltings. “Federated generative privacy”. In: *IEEE Intell. Syst.* 35.4 (2020), pp. 50–57.

[86] Konstantinos Trohidis et al. “Multi-label classification of music into emotions.” In: *ISMIR*. Vol. 8. 2008, pp. 325–330.

[87] Grigorios Tsoumakas and Ioannis Katakis. “Multi-label classification: An overview”. In: *International Journal of Data Warehousing and Mining (IJDWM)* 3.3 (2007), pp. 1–13.

[88] George Tzanetakis and Perry Cook. “Musical genre classification of audio signals”. In: *IEEE Transactions on speech and audio processing* 10.5 (2002), pp. 293–302.

[89] Guile Wu and Shaogang Gong. “Decentralised Learning from Independent Multi-Domain Labels for Person Re-Identification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 4. 2021, pp. 2898–2906.

[90] Andrew C. Yao. “Protocols for secure computations”. In: *23rd annual symposium on foundations of computer science (SFCS 1982)*. IEEE. 1982, pp. 160–164.

[91] Changsong Yu et al. “Multi-level attention model for weakly supervised audio classification”. In: *CoRR* abs/1803.02353 (2018).

[92] Tong Zhang and CCJ Kuo. “Hierarchical classification of audio data for archiving and retrieving”. In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99*. Vol. 6. IEEE. 1999, pp. 3001–3004.

[93] Shuheng Zhou, Katrina Ligett, and Larry Wasserman. “Differential privacy with compression”. In: *2009 IEEE International Symposium on Information Theory*. IEEE. 2009, pp. 2718–2722.

[94] Yuqing Zhu et al. “Private-knn: Practical differential privacy for computer vision”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2020, pp. 11854–11862.

Appendix

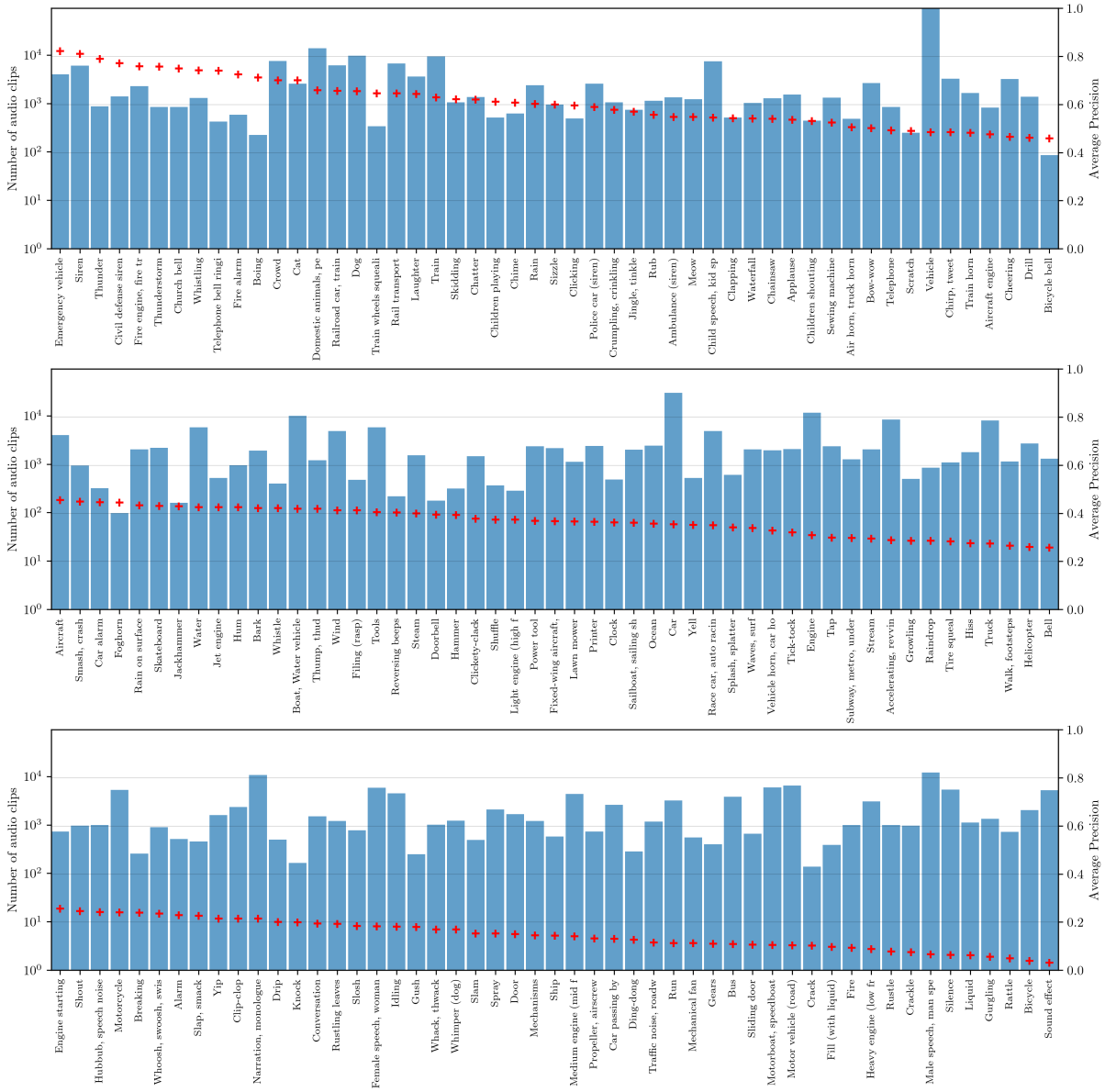


Figure A.1: Average precision per class with amount of training samples after iteration 30.000 for the "big baseline" experiment.

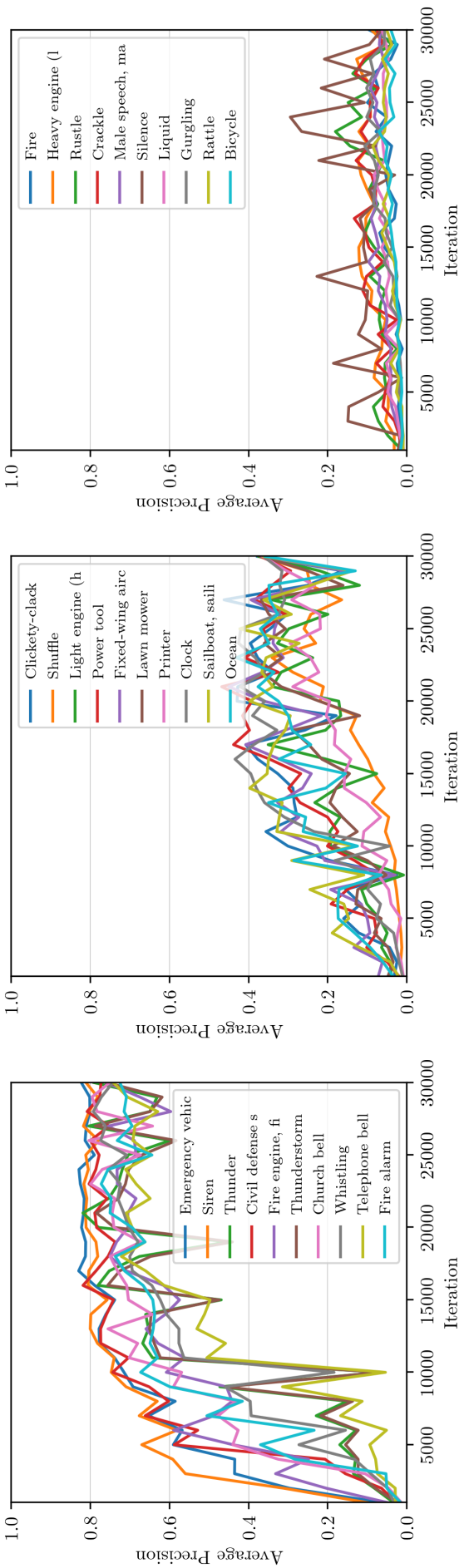


Figure A.2: Average precision per class over iteration for data set "big baseline". On the left ten best classes, ten medium performing in the middle, and ten worst performing classes on the right.

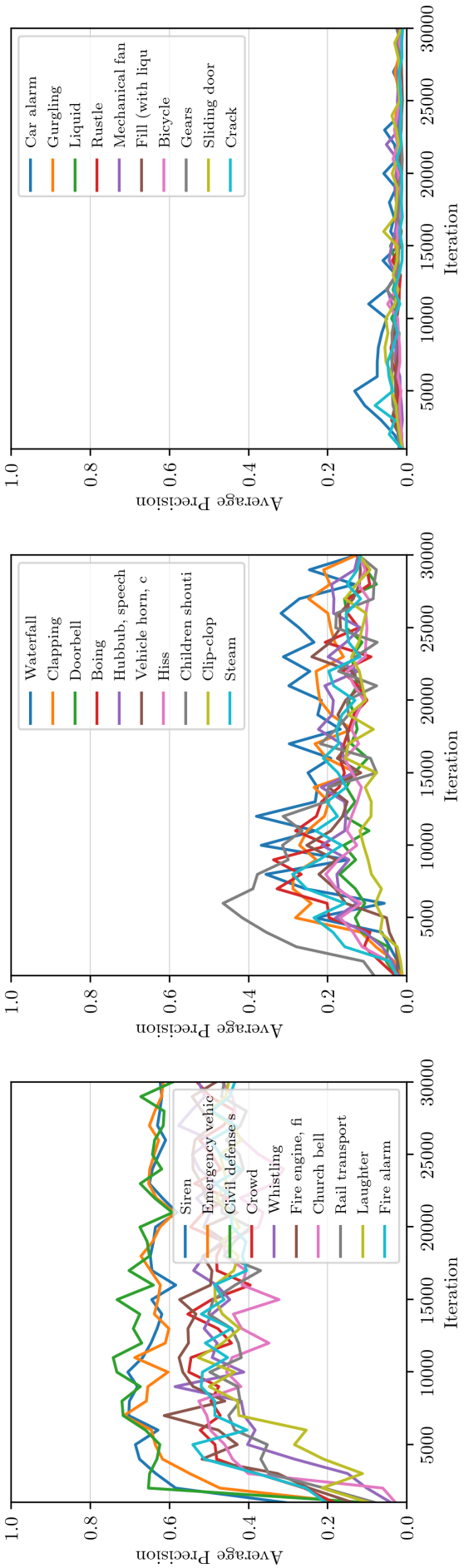


Figure A.3: Average precision per class over iteration for data set "big 10". On the left ten best classes, ten medium performing in the middle, and ten worst performing classes on the right.

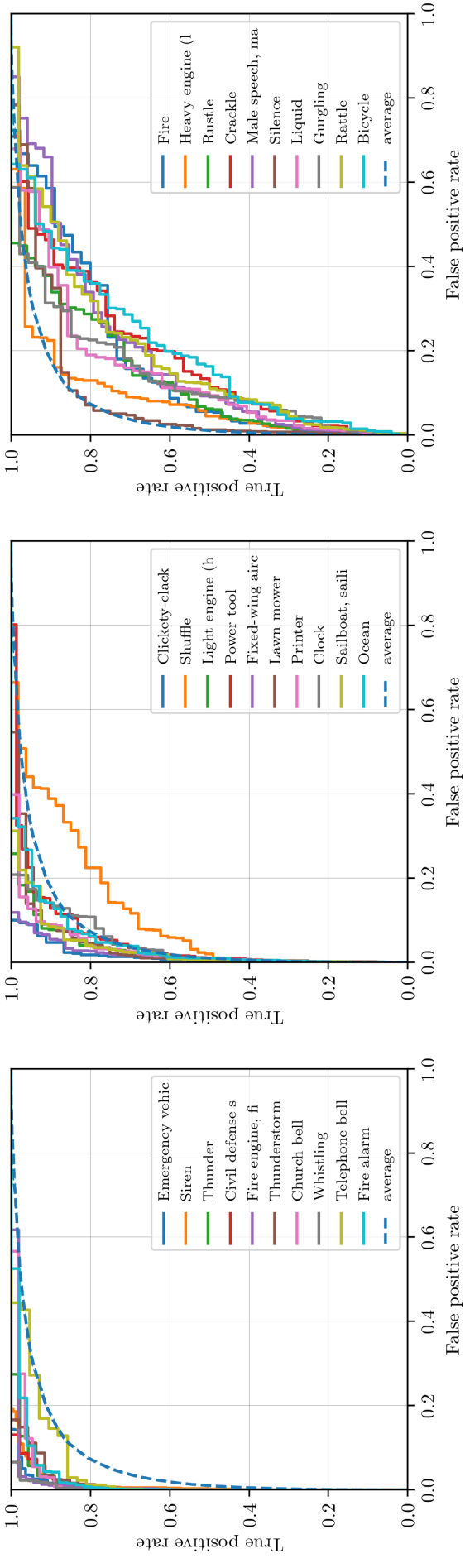


Figure A.4: ROC per class over iteration for the "big baseline". On the left, ten best classes, ten medium performing in the middle, and ten worst performing classes on the right. Dashed the micro-averaged ROC curve for all classes.

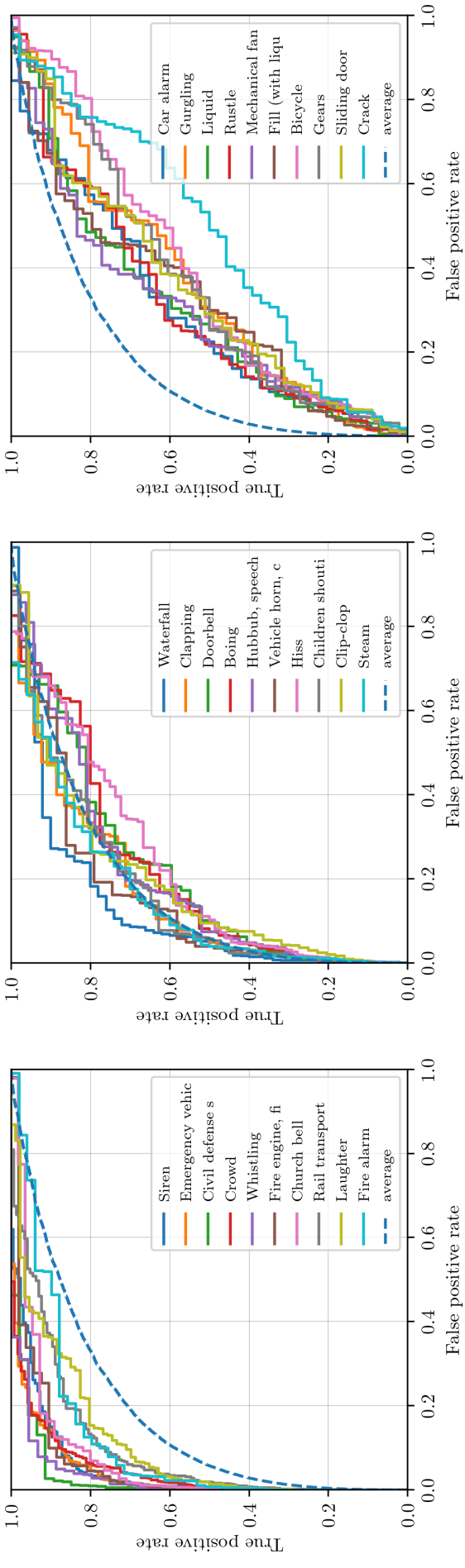


Figure A.5: ROC per class over iteration for data set "big 10". On the left, ten best classes, ten medium performing in the middle, and ten worst performing classes on the right. Dashed the micro-averaged ROC curve for all classes.

#teacher	mAP	AUC	Best performance		individual teacher (avg.)
			thresh.	P / R / F1	P / R / F1
baseline	0.486	0.895	0.3	0.467 / 0.509 / 0.464	n/a
10	0.475	0.900	0.3	0.550 / 0.410 / 0.428	0.412 / 0.389 / 0.359
20	0.446	0.887	0.2	0.398 / 0.517 / 0.437	0.312 / 0.431 / 0.336
50	0.415	0.875	0.2	0.464 / 0.412 / 0.376	0.275 / 0.348 / 0.284

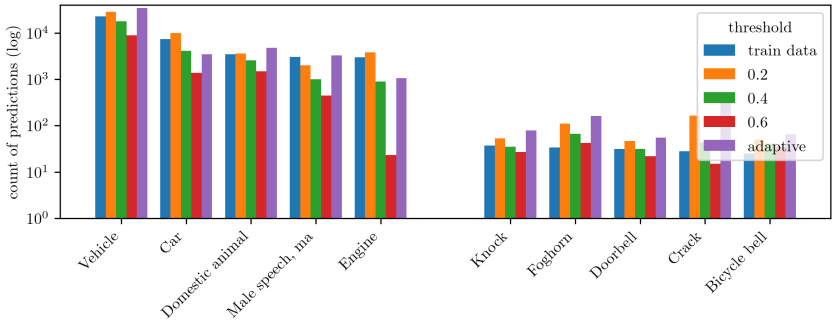
Table A.1: Best performance of teacher ensemble compared to averaged performance of the individual teachers on the "small" data set. No noise applied $\sigma = 0$.

#teachers	threshold	P	R	F1
baseline	adaptive	0.405	0.569	0.457
10	adaptive	0.367	0.570	0.427
20	adaptive	0.359	0.547	0.409
50	adaptive	0.469	0.424	0.361
baseline	0.2	0.559	0.392	0.420
10	0.2	0.404	0.569	0.459
20	0.2	0.398	0.518	0.437
50	0.2	0.464	0.412	0.376
baseline	0.4	0.514	0.440	0.442
10	0.4	0.650	0.300	0.349
20	0.4	0.721	0.251	0.306
50	0.4	0.891	0.219	0.268
baseline	0.6	0.611	0.344	0.389
10	0.6	0.916	0.193	0.241
20	0.6	0.947	0.150	0.194
50	0.6	0.976	0.111	0.155

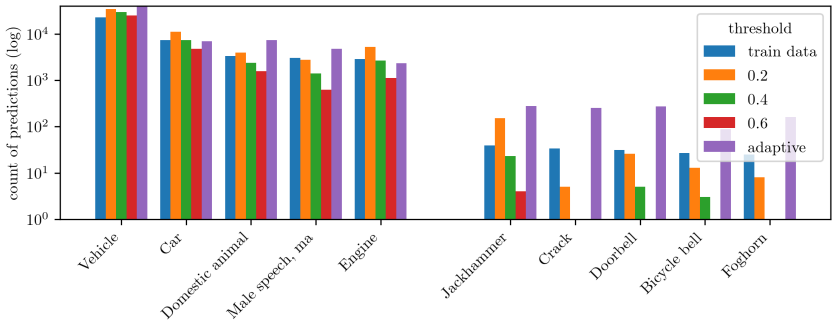
Table A.2: Performance of teacher ensemble for different thresholds and number of teachers on the "small" dataset. No noise applied $\sigma = 0$.

#teachers	threshold	confidence threshold 0.4		confidence threshold 0.6	
		#answ.	P / R / F1	#answ.	P / R / F1
10	adaptive	36%	0.592 / 0.609 / 0.548	14%	0.789 / 0.628 / 0.612
20	adaptive	30%	0.582 / 0.582 / 0.556	12%	0.862 / 0.636 / 0.629
50	adaptive	37%	0.770 / 0.407 / 0.4109	10%	0.949 / 0.581 / 0.655
10	0.2	36%	0.523 / 0.699 / 0.568	14%	0.727 / 0.704 / 0.621
20	0.2	30%	0.500 / 0.658 / 0.548	12%	0.782 / 0.677 / 0.624
50	0.2	37%	0.532 / 0.450 / 0.433	10%	0.855 / 0.623 / 0.616
10	0.4	n/a		14%	0.792 / 0.615 / 0.600
20	0.4	n/a		12%	0.856 / 0.616 / 0.608
50	0.4	n/a		10%	0.954 / 0.575 / 0.649

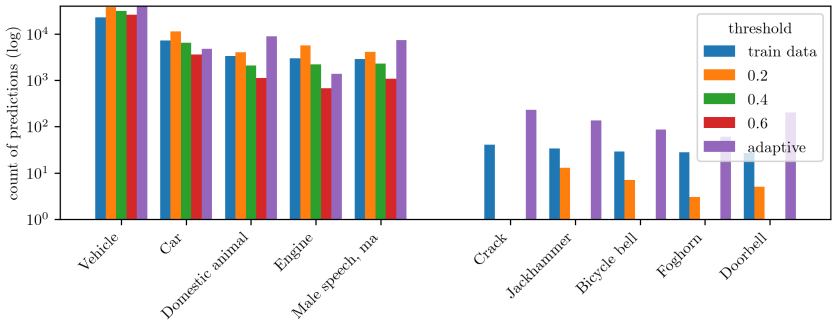
Table A.3: Performance of teacher ensemble for different confidence levels and number of teachers on the "small" dataset. No noise applied $\sigma = 0$.



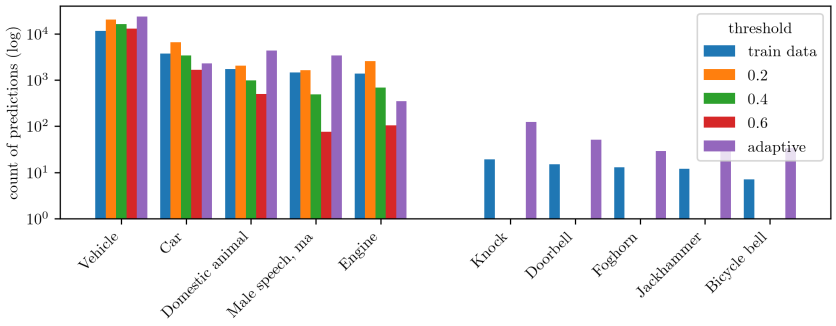
(a) “baseline” teacher ensemble



(b) “big 10” teacher ensemble

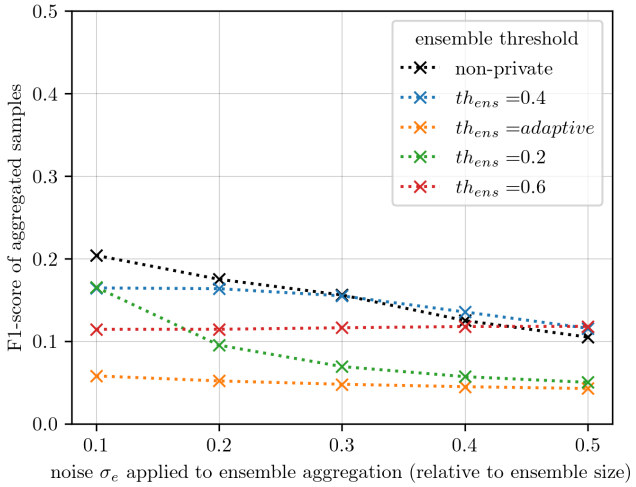


(c) “big 20” teacher ensemble

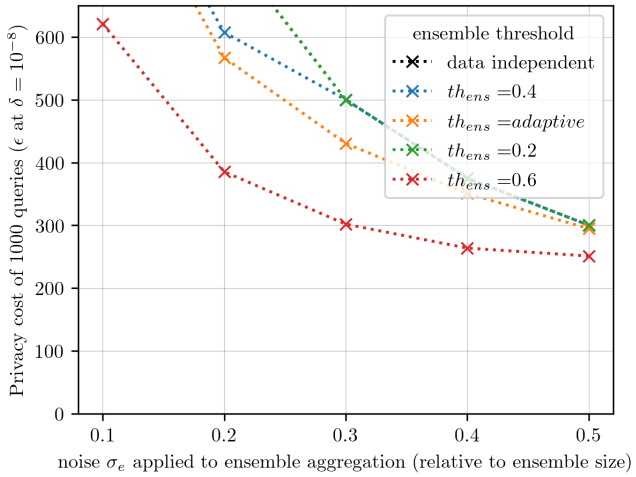


(d) “big 50” teacher ensemble

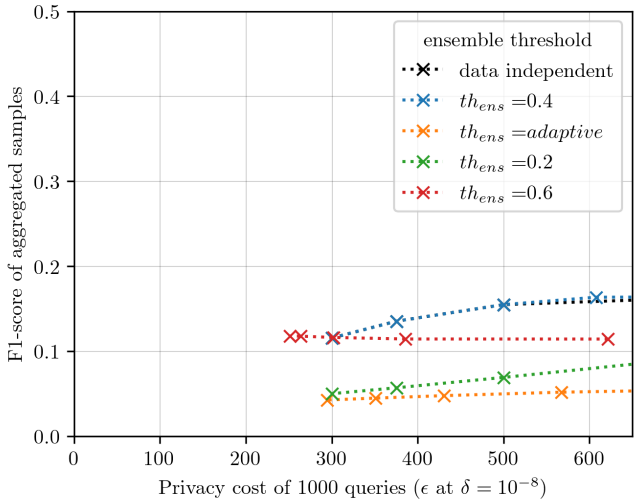
Figure A.6: Count of predictions for selected classes by the ensembles compared to the number of label assignments in the student training set.



(a) F1-performance

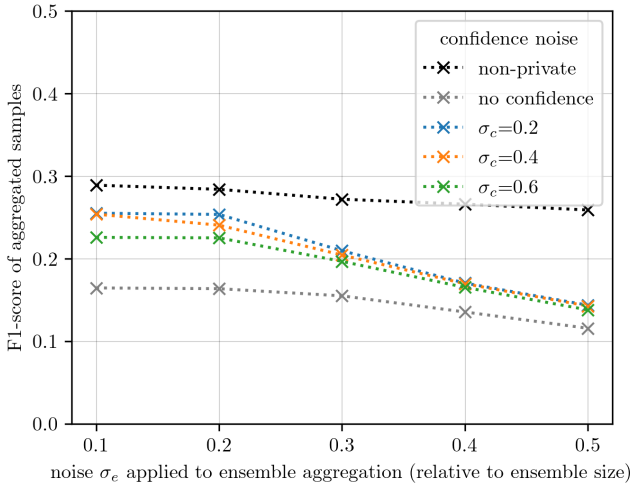


(b) privacy consumption

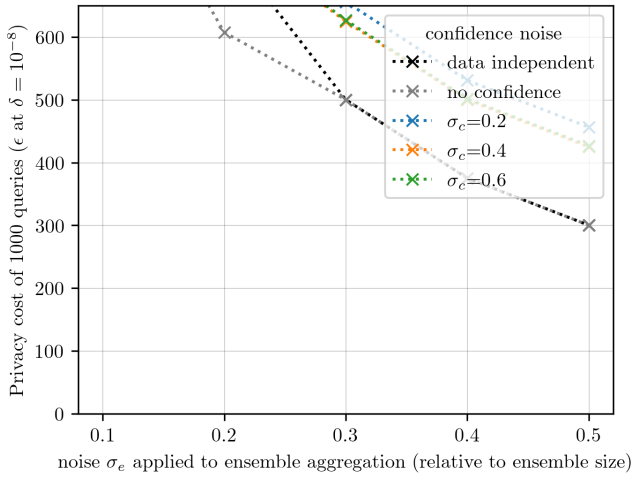


(c) combination

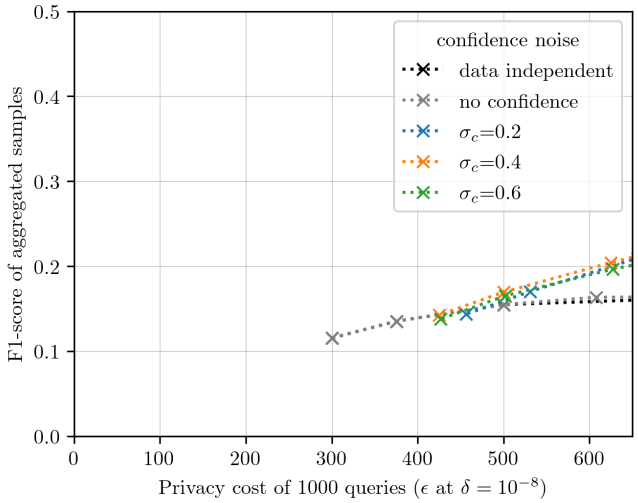
Figure A.7: Ensemble performance for different noise scales and thresholds on the "big 20" experiment.



(a) F1-performance

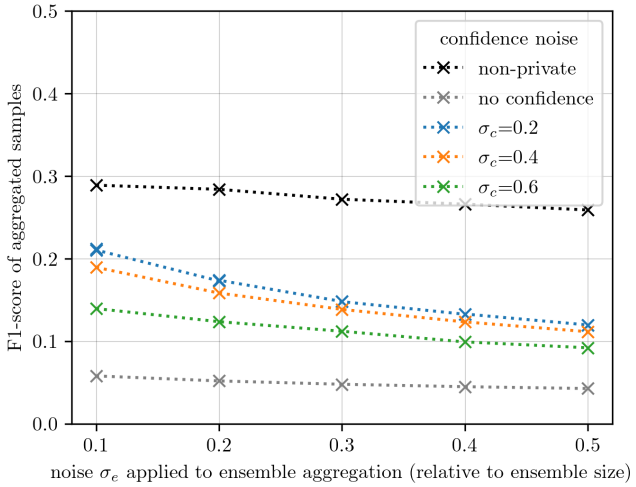


(b) privacy consumption

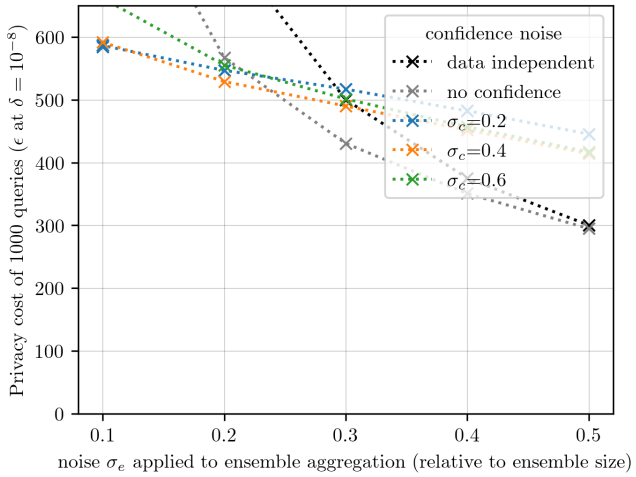


(c) combination

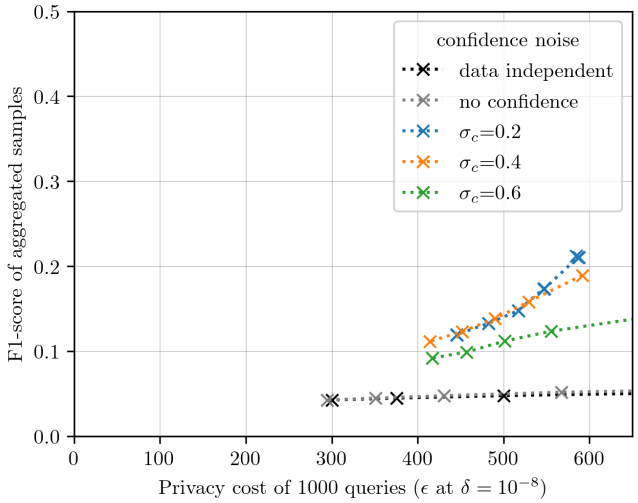
Figure A.8: Ensemble performance for different confidence noise scales on the "big 20" experiment with $th_{ens} = 0.4$ and $th_{conf} = 0.6$.



(a) F1-performance

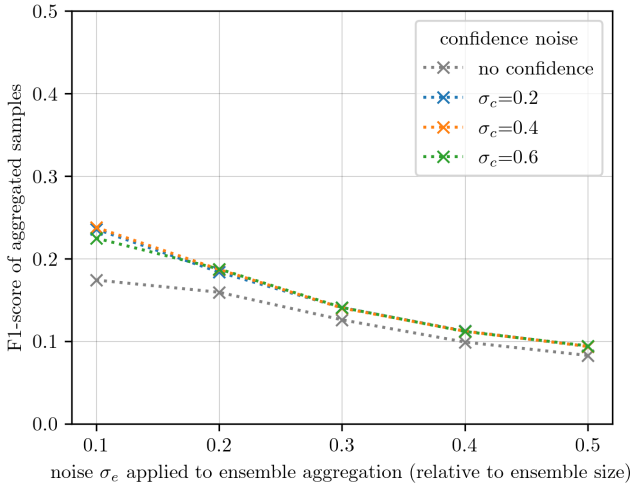


(b) privacy consumption

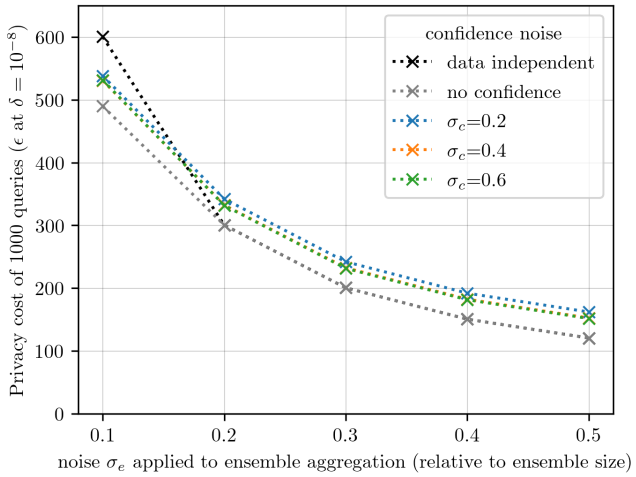


(c) combination

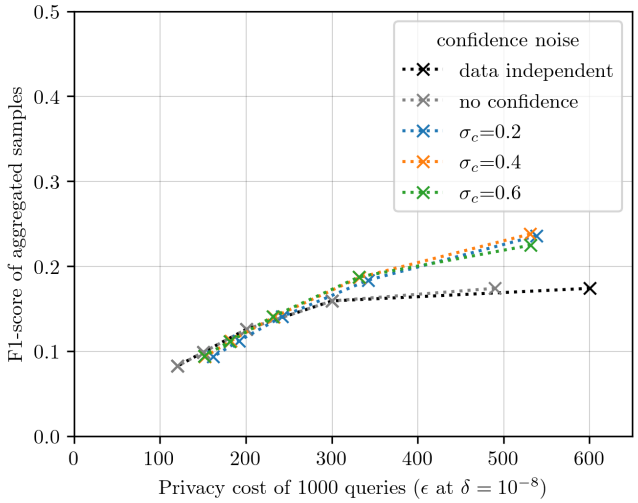
Figure A.9: Ensemble performance for different confidence noise scales on the "big 20" experiment with adaptive ensemble threshold and $th_{conf} = 0.6$.



(a) F1-performance

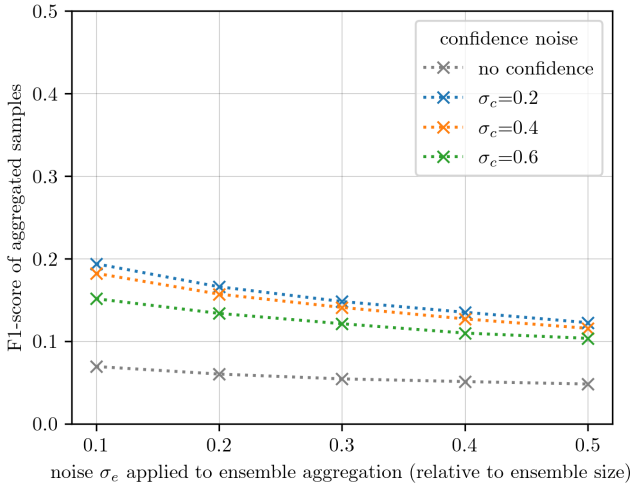


(b) privacy consumption

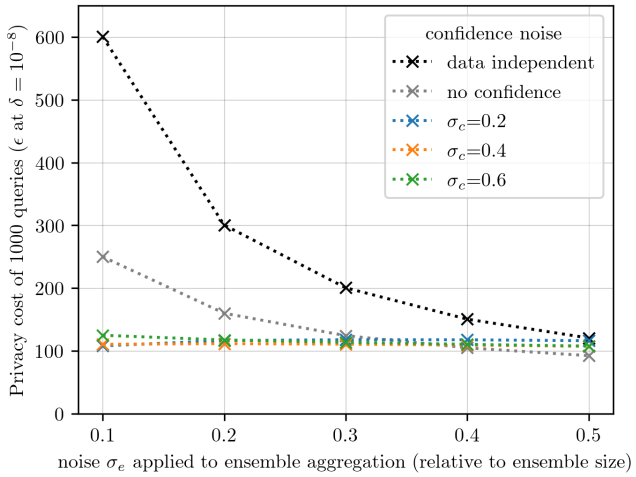


(c) combination

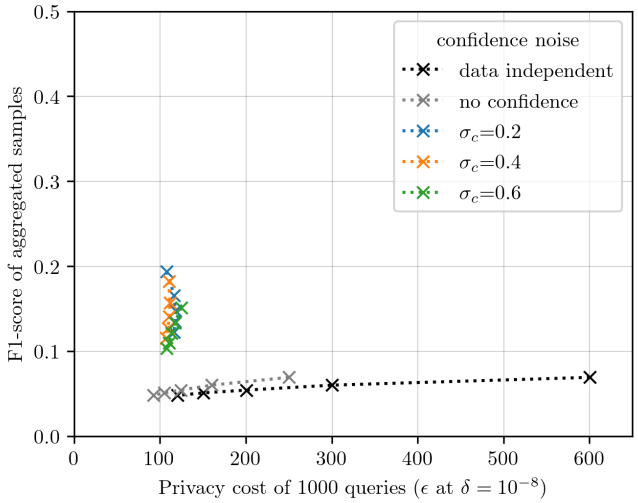
Figure A.10: Ensemble performance for different confidence noise scales on the "big 50" experiment with $th_{ens} = 0.4$ and $th_{conf} = 0.2$.



(a) F1-performance

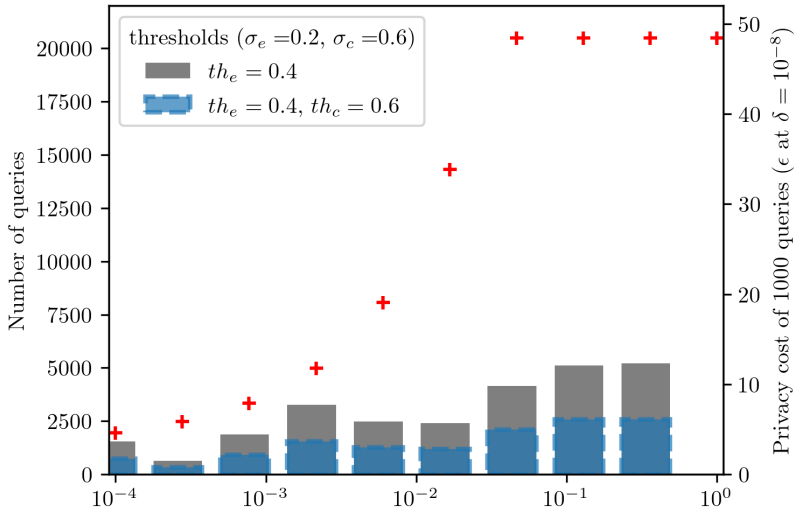


(b) privacy consumption

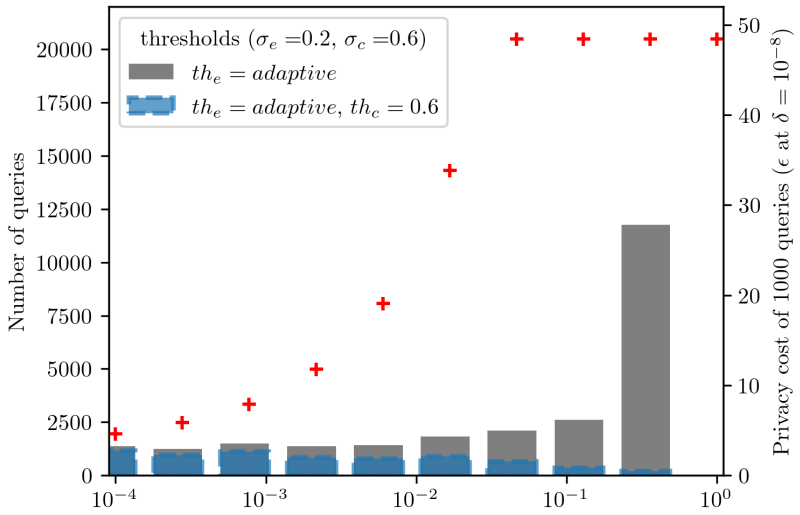


(c) combination

Figure A.11: Ensemble performance for different confidence noise scales on the "big 50" experiment with adaptive ensemble threshold and $th_{conf} = 0.4$.

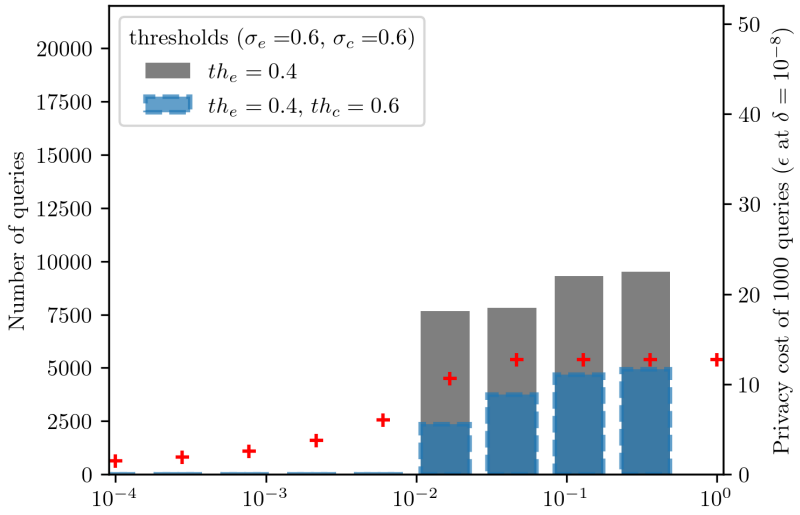


(a) $th_{ens} = 0.4$

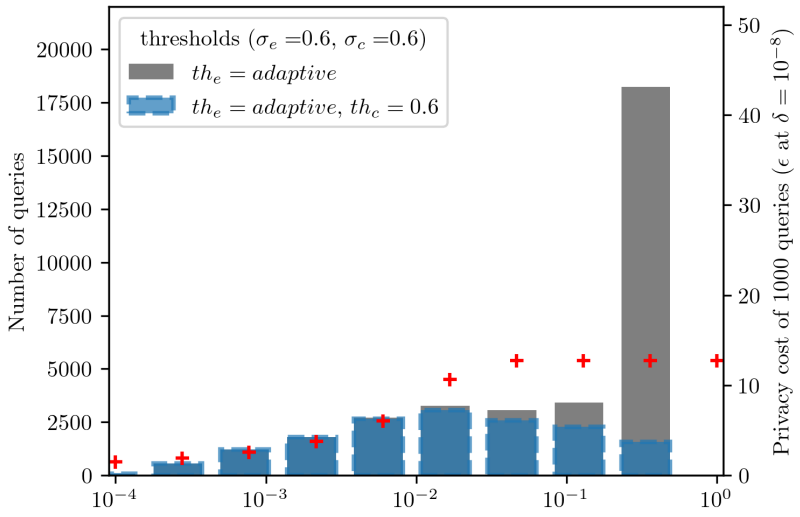


(b) th_{ens} =adaptive

Figure A.12: Distribution of q on the "big 50" predictions with averaged combined privacy cost for the confidence mechanisms ($\sigma_e = 0.2$).



(a) $th_{ens} = 0.4$



(b) th_{ens} =adaptive

Figure A.13: Distribution of q on the "big 50" predictions with averaged combined privacy cost for the confidence mechanisms ($\sigma_e = 0.6$).

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann. Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

Leipzig, December 20, 2021, ...

Paul Muschiol