# PANORAMIA: Privacy Auditing of Machine Learning Models without Retraining

by

Mishaal Kazmi

BSc Computer Science, Lahore University of Management Sciences, 2019

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

**Master of Science**

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

September 2024

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

**PANORAMIA: Privacy Auditing of Machine Learning Models without Retraining**

submitted by **Mishaal Kazmi** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

**Examining Committee:**

Mathias Lécuyer, Assistant Professor, Computer Science, UBC
*Supervisor*

Ivan Beschastnikh, Associate Professor, Computer Science,UBC
*Co-Supervisor*

Thomas Pasquier, Assistant Professor, Computer Science,UBC
*Supervisory Committee Member*

# Abstract

We present PANORAMIA, a privacy leakage measurement framework for machine learning models that relies on membership inference attacks using generated data as non-members. By relying on generated non-member data, PANORAMIA eliminates the common dependency of privacy measurement tools on in-distribution non-member data. As a result, PANORAMIA does not modify the model, training data, or training process, and only requires access to a subset of the training data. We evaluate PANORAMIA on ML models for image and tabular data classification, as well as on large-scale language models. The theory we develop in this paper provides a meaningful step towards addressing privacy measurements in this setting and provides a more rigorous approach to privacy benchmarks for such models. We demonstrate that PANORAMIA's privacy measurements can also be empirically valuable, for instance for providing improved measurements with more data.

# Lay Summary

Machine learning (ML) models power many tasks in modern society but raise concerns about privacy and security, especially when trained on sensitive data like personal, medical, or financial information. This creates risks of data breaches and misuse. Privacy auditing helps assess whether a model reveals private details from its training data, but existing methods often require retraining the model or accessing the full original dataset. PANORAMIA is a framework that addresses these limitations by generating synthetic data resembling the real training data. This artificial data allows privacy leakage measurements without needing non-member real data. PANORAMIA can accurately measure privacy risks in various ML models and datasets, offering valuable insights for improving privacy protection.

# Preface

This thesis presents the original work done by the author, Mishaal Kazmi, conducted in the Systopia lab at the Univerisity of British Columbia under the supervision of Prof. Mathias Lécuyer and Prof. Ivan Beschastnikh. All chapters are adapted from our team's arxiv preprint [16].

This work was performed in collaboration with Hadrien Lautraite, Alireza Akbari, Qiaoyue Tang, Prof. Mathias Lécuyer, and Prof. Sébastien Gambs. In this work, I developed, improved, and implemented the proposed algorithms, and experiments on the image data modality as well as comparison with the baselines. Prof. Mathias Lécuyer devised the idea, framework, and theory behind `PANORAMIA`. Profs. Mathias Lécuyer and Sébastien Gambs suggested framework design and contributed to the paper writing. Profs. Mathias Lécuyer and Ivan Beschastnikh provided feedback, a review of the manuscript, and helpful suggestions for improving paper quality throughout the course of this work. Hadrien Lautraite, Alireza Akbari, worked on the tabular and NLP data modalities respectively. Qiaoyue Tang trained and optimized the DP target models and contributed to refining and extending the theory of this paper.

# Table of Contents

# List of Tables

# List of Figures

# List of Notations

| | |
|---|---|
| $b$ | baseline classifier for $D_{\text{in}}$ vs. $D_{\text{out}}$ |
| $\mathscr{D}$ | distribution over auditor samples |
| $D_f$ | (subset of the) training set of the target model $f$ from $\mathscr{D}$ |
| $D_G$ | training set of the generative model $\mathscr{G}$, with $D_G \subset D_f$ |
| $D_{\text{in}}$ | member auditing set, with $D_{\text{in}} \subset D_f$ and $D_{\text{in}} \cap D_G = \{\}$ |
| $D_{\text{out}}$ | non-member auditing set, with $D_{\text{out}} \sim \mathscr{G}$ |
| $D^{\{\text{tr,te}\}}_{\{\text{in,out}\}}$ | training and testing splits of $D_{\text{in}}$ and $D_{\text{out}}$ |
| $f$ | the target model to be audited. |
| $\mathscr{G}$ | the generative model |
| $m$ | $\lvert D_{\text{in}}\rvert = \lvert D_{\text{out}}\rvert \triangleq m$ |

# Acknowledgments

We are grateful for the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [reference number RGPIN-2022-04469], as well as the Google Research Scholar award. This research was enabled by computational support provided by the Digital Research Alliance of Canada (alliancecan.ca). This research is supported in part by the NSERC Discovery Grant RGPIN-2014-04870 and the Cascadia grant.

I would like to express my sincerest gratitude and appreciation to my supervisors, Prof. Mathias Lécuyer and Prof. Ivan Beschastnikh, both of whom provided invaluable insights, feedback, support, and encouragement to accomplish this goal, paper, and my degree. I am entirely grateful for all their time and support throughout, and without them, none of this would have been achievable. I would also like to extend my Thanks to Prof. Sébastien Gambs for his feedback on my project and his involvement in this work.

I am also extremely grateful to my parents, family, and friends for their undying love and support throughout. A few special shoutouts to my sisters and friends Minahil Kazmi, Fareeha Idrees, Zainab Wattoo and Shadab Shaikh. Last, and most importantly, Thank God for making this all possible.

# Chapter 1

# Introduction

Training Machine Learning (ML) models with Differential Privacy (DP) [9], such as with DP-SGD [1], upper-bounds the worst-case privacy loss incurred by the training data. In contrast, privacy auditing aims to empirically lower-bound the privacy loss of a target ML model or algorithm. In practice, privacy audits usually rely on the link between DP and the performance of membership inference attacks (MIA) [8, 14, 31]. At a high level, DP implies an upper-bound on the performance of MIAs, thus creating a high-performance MIA implies a lower-bound on the privacy loss. Auditing schemes have proven valuable in many settings, such as to audit DP implementations [26], or to study the tightness of DP algorithms [21, 25, 30]. Typical privacy audits rely on retraining the model several times, each time guessing the membership of one sample [6, 12, 35], which is computationally prohibitive, requires access to the target model (entire) training data as well as control over the training pipeline.

To circumvent these concerns, [30] proposed an auditing recipe (called O(1)) requiring only one training run (which could be the same as the actual training) by randomly including/excluding several samples (called auditing examples) into the training dataset of the target model. Later, the membership of the auditing examples are guessed for privacy audit. However, O(1) faces a few challenges in certain setups. First, canaries, which are datapoints specially crafted to be easy to detect when added to the training set [21, 26, 30], cannot be employed as auditing examples when measuring the privacy leakage for data that a contributor actually puts into the

model, and not a *worst case data point*. This matches a setting in which individual data contributors (*e.g.*, a hospital in a cross-site Federated Learning (FL) setting or a user of a service that trains ML models on users' data) measure the leakage of their own (*i.e.*, known) partial training data in the final trained model. Second, O(1) also relies on the withdrawal of real data from the model to construct non-member in-distribution data. This is problematic in situations in which ML model owners need to conduct post-hoc audits, in which case it is too late for removal [27]. Moreover, in-distribution audits require much more data, thus withholding many data points (typically more than the test set size) and reducing model utility. This brings us to the question: *Given an instance of a machine learning model as a target, can we perform post-hoc estimation of the privacy loss with regards to a known member subset of the target model training dataset?*

**Our contributions.** We propose PANORAMIA, a new scheme for *Privacy Auditing with NO Retraining by using Artificial data for Membership Inference Attacks*. More precisely, we consider an auditor with access to a subset of the training data and introduce a new alternative for accessing non-members: using synthetic datapoints from a generative model trained on the member data, unlocking the limit on non-member data. PANORAMIA uses this generated data, together with known members, to train and evaluate a MIA attack on the target model to audit (§4). We also adapt the theory of privacy audits, and show how PANORAMIA can estimate the privacy loss (though not a lower-bound) of the target model with regards to the known member subset (§5). An important benefit of PANORAMIA is to perform privacy loss measurements with (1) no retraining the target ML model (*i.e.*, we audit the end-model, not the training algorithm), (2) no alteration of the model, dataset, or training procedure, and (3) only partial knowledge of the training set. We evaluate PANORAMIA on CIFAR10 models and observe that overfitted models, larger models, and models with larger DP parameters have higher measured privacy leakage. We also demonstrate the applicability of our approach on the GPT-2 based model (*i.e.*, WikiText dataset) and CelebA models.

# Chapter 2

# Background

DP is the established privacy definition in the context of ML models, as well as for data analysis in general. We focus on the pure DP definition to quantify privacy loss with well-understood semantics. In a nutshell, DP is a property of a randomized mechanism (or computation) from datasets to an output space $\mathscr{O}$, noted $M : \mathscr{D} \to \mathscr{O}$. It is defined over neighboring datasets $D, D'$, differing by one element $x \in \mathscr{X}$ (we use the add/remove neighboring definition), which is $D' = D \cup \{x\}$. Formally:

**Definition 1** (Differential Privacy [9])**.** *A mechanism $M : \mathscr{D} \to \mathscr{O}$ is $\varepsilon$-DP if for any two neighbouring datasets $D$, $D' \in \mathscr{D}$, and for any measurable output subset $O \subseteq \mathscr{O}$ it holds that:*

$$P[M(D) \in O] \leq e^{\varepsilon} P[M(D') \in O].$$

Since the neighbouring definition is symmetric, so is the DP definition, and we also have that $P[M(D') \in O] \leq e^{\varepsilon} P[M(D) \in O]$. Intuitively, $\varepsilon$ upper-bounds the worst-case contribution of any individual example to the distribution over outputs of the computation (*i.e.*, the ML model learned). More formally, $\varepsilon$ is an upper-bound on the *privacy loss* incurred by observing an output $o$, defined as $\left| \ln \left( \frac{\mathbb{P}[M(D)=o]}{\mathbb{P}[M(D')=o]} \right) \right|$, which quantifies how much an adversary can learn to distinguish $D$ and $D'$ based on observing output $o$ from $M$. A smaller $\varepsilon$ hence means higher privacy.

**DP, MIA and privacy audits.** To audit a DP training algorithm $M$ that outputs a model $f$, one can perform a MIA on datapoint $x$, trying to distinguish between a neighboring training sets $D$ and $D' = D \cup \{x\}$. The MIA can be formalized as

a hypothesis test to distinguish between $\mathscr{H}_0 = D$ and $\mathscr{H}_1 = D'$ using the output of the computation $f$. Dong et al. [8], Kairouz et al. [14], Wasserman and Zhou [31] show that any such test at significance level $\alpha$ (False Positive Rate or FPR) has power (True Positive Rate or TPR) bounded by $e^{\varepsilon}\alpha$. In practice, one repeats the process of training model $f$ with and without $x$ in the training set, and uses a MIA to guess whether $x$ was included. If the MIA has TPR $> e^{\varepsilon}$FPR, the training procedure that outputs $f$ is not $\varepsilon$-DP. This is the building block of most privacy audits [12, 21, 25, 26, 35].

**Averaging over data instead of models with O(1).** The above result bounds the success rate of MIAs when performed over several *retrained models*, on two alternative datasets $D$ and $D'$. Steinke et al. [30] show that it is possible to average *over data* when several data points independently differ between $D$ and $D'$. Let $x_{1,m}$ be the $m$ data points independently included in the training set, and $s_{1,m} \in \{0,1\}^m$ be the vector encoding inclusion. $T_{0,m} \in \mathbb{R}^m$ represents any vector of guesses, with positive values for inclusion in the training set (member), negative values for non-member, and zero for abstaining. Then, if the training procedure is $\varepsilon$-DP, Proposition 5.1 in Steinke et al. [30] bounds the performance of guesses from $T$ with:

$$\mathbb{P}\Big[\sum_{i=1}^{m} \max\{0, T_i \cdot S_i\} \geq v \mid T = t\Big] \leq \mathop{\mathbb{P}}_{S' \sim \text{Bernoulli}(\frac{e^{\varepsilon}}{1+e^{\varepsilon}})^m}\Big[\sum_{i=1}^{m} |t_i| \cdot S'_i \geq v\Big].$$

In other words, an audit (MIA) $T$ that can guess membership better than a Bernoulli random variable with probability $\frac{e^{\varepsilon}}{1+e^{\varepsilon}}$ refutes an $\varepsilon$-DP claim. In this work we build on this result, extending the algorithm (§4) and theoretical analysis (§5) to enable the use of generated data for non-members. The key difference compared to our work lies in how we create the audit set. In Steinke et al. [30], the audit set is fixed, and data points are randomly assigned to member or non-member by a Bernoulli random variable $S$. Members are actually used in training the target model $f$, while non-members are not (so assignment happens before training). In our framework, we take a set of known iid. members (after the fact), and pair each point with a non-member (generated iid. from the generator distribution). We then flip $S$ to sample which one will be shown to the "auditor" (MIA/baseline) for testing, thereby creating the test task of our privacy measurement.

# Chapter 3

# Related work

Recent privacy auditing work measures the privacy of an ML model by lower-bounding its privacy loss. This usually requires altering the training pipeline of the ML model, either by injecting canaries that act as outliers [4] or by using data poisoning attack mechanisms to search for worst-case memorization [12, 25]. MIAs are also increasingly used in privacy auditing, to estimate the degree of memorization of member data by an ML algorithm by resampling the target algorithm $\mathscr{M}$ to bound $\frac{P(M|in)}{P(M|out)}$ [13].

The auditing procedure usually involves searching for optimal neighboring datasets $D, D'$ and sampling the DP outputs $\mathscr{M}(D), \mathscr{M}(D')$, to get a Monte Carlo estimate of $\varepsilon$. This approach raises important challenges. First, existing search methods for neighboring inputs, involving enumeration or symbolic search, are impossible to scale to large datasets, making it difficult to find optimal dataset pairs. In addition, Monte Carlo estimation requires up to thousands of costly model retrainings to bound $\varepsilon$ with high confidence. Consequently, existing approaches for auditing ML models predominantly require the re-training of ML models for every (batch of) audit queries, which is computationally expensive in large-scale systems [12, 21, 35].

This makes privacy auditing computationally expensive and gives an estimate by averaging over models, which might not reflect the true guarantee of a specific pipeline deployed in practice.

Nonetheless, improvements to auditing have been made in a variety of directions.

For example, Nasr et al. [26] and Maddock et al. [22] have taken advantage of the iterative nature of DP-SGD, auditing individual steps to understand the privacy of the end-to-end algorithm. The work by Andrew et al. [2] leverages the fact that analyzing MIAs for non-member data does not require re-running the algorithm. Instead, it is possible to re-sample the non-member data point: if the data points are i.i.d. from an asymptotically Gaussian distribution with mean zero and variance $1/d$, this enables a closed-form analysis of the non-member case.

Recently, the authors of Steinke et al. [30] proposed a novel scheme for auditing differential privacy with $O(1)$ training rounds. This approach enables privacy audits using multiple training examples from the same model training, if examples are included in training independently (which requires control over the training phase, and altering the target model). They demonstrate the effectiveness of this new approach on DP-SGD, in which they achieve meaningful empirical privacy lower bounds by training only one model, whereas standard methods would require training hundreds of models. The key difference compared to our work lies in how we create the audit set. In the paper by Steinke et al. [30], the audit set is fixed, and data points are randomly assigned to member or non-member by a Bernoulli random variable $S$. Members are actually used in training the target model $f$, while non-members are not (so assignment happens before training). In our framework, we take a set of known iid. members (after the fact), and pair each point with a non-member(generated iid. from the generator distribution). We then flip $S$ to sample which one will be shown to the "auditor" (MIA/baseline) for testing, thereby creating the test task of our privacy measurement.

# Chapter 4

# PANORAMIA design

In this chapter, we describe in detail the design for PANORAMIA: *Privacy Auditing with NO Retraining using Artificial data for Membership Inference Attacks*. PANORAMIA is our privacy leakage measurement framework that allows us to assess ML models for their privacy leakage on the data they are trained on, using synthetic data as non-members.

Figure 4.1 summarizes the end-to-end PANORAMIA privacy measurement. The measurement starts with a target model $f$, and a subset of its training data $D_f$ from distribution $\mathscr{D}$. For instance, $\mathscr{D}$ and $D_f$ could be the distribution and dataset of one participant in an FL training procedure that outputs a final model $f$. The privacy measurement then proceeds in two phases. **Phase 1:** In the first phase, PANORAMIA uses a subset of the known training data $D_G \subset D_f$ to train a generative model $\mathscr{G}$. The goal of the generative model $\mathscr{G}$ is to match the training data distribution $\mathscr{D}$ as closely as possible, which is formalized in Definition 3 (§5). Using the generative model $\mathscr{G}$, we can synthesize non-member data, which corresponds to data that was not used in the training of target model $f$. Hence, we now have access to an independent dataset of member data $D_{\text{in}} = D_f \backslash D_G$, and a synthesized dataset of non-member data $D_{\text{out}} \sim \mathscr{G}$, of size $m = |D_{\text{in}}|$.

**Phase 2:** In the second phase, we leverage $D_{\text{in}}$ and $D_{\text{out}}$ to audit the privacy leakage of $f$ using a MIA. To this end, we split $D_{\text{in}}, D_{\text{out}}$ into training and testing sets, respectively called $D_{\text{in}}^{\text{tr}}, D_{\text{out}}^{\text{tr}}$ and $D_{\text{in}}^{\text{te}}, D_{\text{out}}^{\text{te}}$. We use the training set to train a MIA (called PANORAMIA in Figure 4.1), a binary classifier that predicts whether

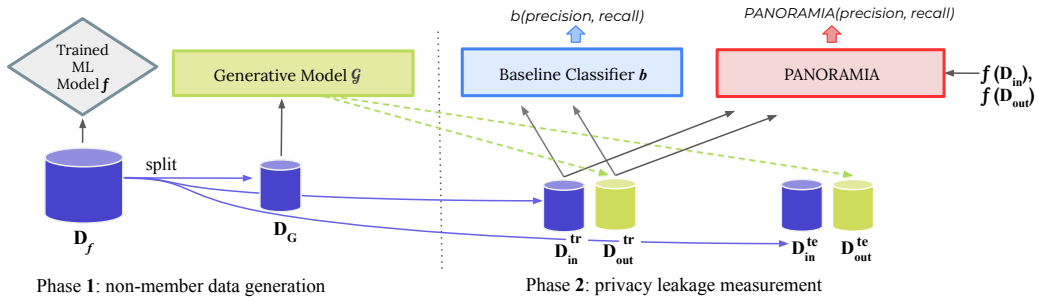



**Figure 4.1:** PANORAMIA's two phases audit. Phase 1: training of generative model $\mathscr{G}$ using member data. Phase 2: training a MIA using member data and generated non-member data. Comparison of its performance to that of a baseline without access to $f$. Notations are found under list of notations.

a given datapoint is a member of $D_f$, the training set of the target model $f$. This MIA classifier makes its prediction based on both a training example $x$, as well as information from applying the target model $f$ to the input, such as the loss of the target model when applied to this example $\text{loss}(f(x))$ (see §8, Chapter 6 for details). We use the test set to measure the MIA performance, using the precision at different recall values. Previous results linking the performance of a MIA on several data-points to $\varepsilon$-DP bounds rely on independence between members and non-members. This intuitively means that there is no information about membership in $x$ itself. When the auditor controls the training process this independence is enforced by construction, by adding data points to the training set based on an independent coin flip.

In PANORAMIA, we do not have independence between membership and $x$, as all non-members come from the generator $\mathscr{G} \neq \mathscr{D}$. As a result, there are two ways to guess membership and have high MIA precision: either by using $f$ to detect membership (*i.e.*, symptomatic of privacy leakage) or by detecting generated data (*i.e.*, not a symptom of privacy leakage). To measure the privacy leakage, we compare the results of the MIA to that of a baseline classifier $b$ that guesses membership based exclusively on $x$, without access to $f$. The stronger this baseline, the better the removal of the effect of synthesized data detection. Algorithm 1

**Algorithm 1** PANORAMIA

**Input:** Target ML model $f$, audit set size $m$, confidence $1-\beta$

**Phase 1:**

1: Split $D_f$ in $D_G, D^{\text{tr}}_{\text{in}}, D^{\text{te}}_{\text{in}}$, with $|D^{\text{te}}_{\text{in}}| = m$
2: Train generator $\mathscr{G}$ on $D_G$
3: Generate $D^{\text{tr}}_{\text{out}}, D^{\text{te}}_{\text{out}}$ of size $|D^{\text{tr}}_{\text{in}}|, |D^{\text{te}}_{\text{in}}|$

**Phase 2:**

Train the baseline and MIA:

1: Label $D^{\text{tr}}_{\text{in}}$ as members, and $D^{\text{tr}}_{\text{out}}$ as non-members
2: Train $b$ to predict labels using $x \in D^{\text{tr}}_{\text{in}} \cup D^{\text{tr}}_{\text{out}}$
3: Train MIA to predict labels using $x \in D^{\text{tr}}_{\text{in}} \cup D^{\text{tr}}_{\text{out}}$ and $f(x)$

Measure privacy leakage (see §5):

1: Sample $s \sim \text{Bernoulli}(\frac{1}{2})^m$ ▷ Def.2
2: Create audit set $X = s \cdot D^{\text{te}}_{\text{in}} + (1-s) D^{\text{te}}_{\text{out}}$
3: Score each audit point for membership, creating $t^b \triangleq b(X) \in \mathbb{R}^m_+$ and $t^a \triangleq \text{MIA}(X) \in \mathbb{R}^m_+$
4: Set $v^b_{\text{ub}}(c,t) \triangleq \sup\{v : \beta^b(m,c,v,t) \leq \frac{\beta}{2}\}$ ▷ Prop.1
5: $c_{\text{lb}} = \max_{t,c} \mathbb{1}\{t^b \geq t\} \cdot s \leq v^b_{\text{ub}}(c, \mathbb{1}\{t^b \geq t\})$
6: Set $v^a_{\text{ub}}(c,\varepsilon,t) \triangleq \sup\{v : \beta^a(m,c,\varepsilon,v,t) \leq \frac{\beta}{2}\}$ ▷ Prop.2
7: $\{c+\varepsilon\}_{\text{lb}} = \max_{t,c,\varepsilon} \mathbb{1}\{t^a \geq t\} \cdot s \leq v^a_{\text{ub}}(c, \varepsilon, \mathbb{1}\{t^a \geq t\})$

**Return** $\tilde{\varepsilon} \triangleq \{c+\varepsilon\}_{\text{lb}} - c_{\text{lb}}$

summarizes the entire procedure. In the next section, we demonstrate how to relate the difference between the baseline $b$ and the MIA performance to the privacy loss $\varepsilon$.

# Chapter 5

# Formalizing PANORAMIA framework and auditing game

In this chapter, we formalize our theoretical framework and auditing game to quantify privacy leakage via PANORAMIA. The first step to quantifying privacy leakage is to formalize an auditing game. PANORAMIA starts with $x^{\text{in}} \in \mathscr{X}^m$ training points, coming from the training data distribution $\mathscr{D}$ to be audited (*e.g.*, the data distribution of one participant in an FL setting), as well as $x^{\text{gen}} \in \mathscr{X}^m$ generated points, coming from the generator distribution $\mathscr{G}$ ($x^{\text{gen}} \sim \mathscr{G}$). The sequence of auditing samples $x \in \mathscr{X}^m$ is created as follows:

**Definition 2** (Auditing game)**.**

$$s \sim \textit{Bernoulli}(\frac{1}{2})^m, \textit{ with } s_i \in \{0,1\},$$
$$x_i = (1-s_i)x_i^{gen} + s_i x_i^{in}, \ \forall i \in \{1,\ldots,m\}.$$

That is, $s$ is sampled independently to choose either the real ($s_i = 1$) or generated ($s_1 = 0$) data point at each index $i$. This creates a sequence of $m$ examples that PANORAMIA will try to tell apart (*i.e.*, guess $s$). The level of success achievable in this game will quantify the privacy leakage of the target model $f$. We follow an analysis inspired by that of [30], but require several key technical changes to support auditing with no retraining using generated non-member data. The first

major change is the introduction of a new game based on generated data, which requires accounting for the quality of the generator in the hypothesis test and the analysis, and interpreting results accordingly. The second adaptation is to focus on member detection, ignoring non-members. This lets us soften the requirements put on the data generator, which only needs to assign high likelihood to real data, but can (and does) occasionally generate poor samples that are easy to detect.

In what follows, we first formalize PANORAMIA's audit procedure as a hypothesis test on our auditing game, for which we construct a statistical test (§5.1). Then, we show how to use this hypothesis test to quantify privacy leakage as a lower confidence interval, and interpret the semantics of our privacy leakage measurements (§5.1.1).

## 5.1 Formalizing the Audit as a Hypothesis Test

We first need a notion of quality for our generator:

**Definition 3** ($c$-closeness)**.** *For all $c > 0$, we say that a generative model $\mathscr{G}$ is $c$-close for data distribution $\mathscr{D}$ if:*

$$\forall x \in \mathscr{X},\ e^{-c}\mathbb{P}_{\mathscr{D}}\left[x\right] \leq \mathbb{P}_{\mathscr{G}}\left[x\right].$$

The smaller $c$ the better, as it means that generator $\mathscr{G}$ assigns to real data a probability that cannot be too small compared to that of the real data distribution. Notice that our definition of generator closeness is very similar to that of DP. This is not a coincidence as we will use this definition to be able to reject claims of both $c$-closeness for the generator and $\varepsilon$-DP for the target model. We further note that contrary to the DP definition, $c$-closeness is one-sided, as we only bound $\mathbb{P}_{\mathscr{G}}$ from below. Intuitively, this means that the generator has to produce high-quality samples (*i.e.*, samples likely under the data distribution $\mathscr{D}$) with high enough probability. Thus, it does not require that all samples are good, and the generator is allowed to occasionally generate bad samples (that are unlikely under $\mathscr{D}$). This one-sided measure of closeness is enabled by our focus on detecting members (*i.e.*, true positives) as opposed to members and non-members. It is important as

it puts less stringent requirements on the generator, which has to sometimes, but "not always", fool the baseline, while still enabling PANORAMIA's audit with this weaker constraint.

Using this definition, we can formulate the hypothesis test on our auditing game that underpins our approach:

$$\mathscr{H} : \text{generator } \mathscr{G} \text{ is } c\text{-close, and target model } f \text{ is } \varepsilon\text{-DP.}$$

To construct a statistical test allowing us to reject $\mathscr{H}$ based on evidence, we define two key mechanisms (corresponding to PANORAMIA's auditing scheme). First, the (potentially randomized) baseline guessing mechanism $B(s,x) : \{0,1\}^m \times \mathscr{X}^m \to \mathbb{R}_+^m$, which outputs a (non-negative) score for the membership of each datapoint $x_i$, based on this datapoint only. That is, $B(s,x) = \{b(x_1), b(x_2), \ldots, b(x_m)\}$.

Second, we define $A(s,x,f) : \{0,1\}^m \times \mathscr{X}^m \times \mathscr{F} \to \mathbb{R}_+^m$, which outputs a (potentially randomized) non-negative score for the membership of each datapoint, with the guess for index $i$ depending on $x_{\leq i}$ and target model $f$. Note that if the target model $f$ is DP, then $A$ is DP w.r.t. inclusion in the dataset $s$, outside of what is revealed by $x$. We are now ready to construct a hypothesis test for $\mathscr{H}$. First, we construct tests for each part of the hypothesis separately.

**Proposition 1.** *Let $\mathscr{G}$ be c-close, and $T^b \triangleq B(S,X)$ be the guess from the baseline. Then, for all $v \in \mathbb{R}$ and all $t$ in the support of $T$:*

$$\begin{aligned}
&\mathbb{P}_{S,X,T^b}\Big[\sum_{i=1}^{m} T_i^b \cdot S_i \geq v \mid T^b = t^b\Big] \\
&\leq \mathop{\mathbb{P}}_{S' \sim Bernoulli(\frac{e^c}{1+e^c})^m}\Big[\sum_{i=1}^{m} t_i^b \cdot S_i' \geq v\Big] \triangleq \beta^b(m,c,v,t^b)
\end{aligned}$$

*Proof.* Notice that under our baseline model $B(s,x) = \{b(x_1), b(x_2), \ldots, b(x_m)\}$, and given that the $X_i$ are i.i.d., we have that: $S_{<i} \perp\!\!\!\perp T_{<i}^b \mid X_{<i}$, since $T_i^b = B(S,X)_i$'s distribution is entirely determined by $X_i$; and $S_{\leq i} \perp\!\!\!\perp T_{>i}^b \mid X_{<i}$ since the $X_i$ are sampled independently from the past.

We study the distribution of $S$ given a fixed prediction vector $t^b$, one element

$i \in [m]$ at a time:

$$
\begin{aligned}
&\mathbb{P}\big[S_i = 1 \mid T^b = t^b, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&= \mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&= \mathbb{P}\big[X_i \mid S_i = 1, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big] \\
&\quad \frac{\mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]}{\mathbb{P}\big[X_i \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \\
&= \frac{\mathbb{P}\big[X_i \mid S_i = 1, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]\mathbb{P}\big[S_i = 1\big]}{\mathbb{P}\big[X_i \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \\
&= \frac{\mathbb{P}\big[X_i \mid S_i = 1\big]\frac{1}{2}}{\mathbb{P}\big[X_i \mid S_i = 1\big]\frac{1}{2} + \mathbb{P}\big[X_i \mid S_i = 0\big]\frac{1}{2}} \\
&= \frac{1}{1 + \frac{\mathbb{P}\left[X_i \mid S_i = 0\right]}{\mathbb{P}\left[X_i \mid S_i = 1\right]}} = \frac{1}{1 + \frac{\mathbb{P}_{\mathscr{G}}\left[X_i\right]}{\mathbb{P}_{\mathscr{D}}\left[X_i\right]}} \leq \frac{1}{1 + e^{-c}} = \frac{e^c}{1 + e^c}
\end{aligned}
$$

The first equality uses the independence remarks at the beginning of the proof; the second relies on Bayes' rule; while the third and fourth that $S_i$ is sampled i.i.d. from a Bernoulli with probability half, and $X_i$ i.i.d. conditioned on $S_i$. The last inequality uses Definition 3 for $c$-closeness.

Using this result and the law of total probability to introduce conditioning on $X_{\leq i}$, we get that:

$$
\begin{aligned}
&\mathbb{P}\big[S_i = 1 \mid T^b = t^b, S_{<i} = s_{<i}\big] \\
&= \sum_{x_{\leq i}} \mathbb{P}\big[S_i = 1 \mid T^b = t^b, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad \mathbb{P}\big[X_{\leq i} = x_{\leq i} \mid T^b = t^b, S_{<i} = s_{<i}\big] \\
&\leq \sum_{x_{\leq i}} \frac{e^c}{1 + e^c} \mathbb{P}\big[X_{\leq i} = x_{\leq i} \mid T^b = t^b, S_{<i} = s_{<i}\big],
\end{aligned}
$$

and hence that:

$$
\mathbb{P}\big[S_i = 1 \mid T^b = t^b, S_{<i} = s_{<i}\big] \leq \frac{e^c}{1 + e^c} \tag{5.1}
$$

We can now proceed by induction: assume inductively that $W_{m-1} \triangleq \sum_{i=1}^{m-1} T_i^b \cdot S_i$ is stochastically dominated (see Definition 4.8 in [30]) by $W'_{m-1} \triangleq \sum_{i=1}^{m-1} T_i^b \cdot S'_i$, in

which $S' \sim \text{Bernoulli}(\frac{e^c}{1+e^c})^{m-1}$. Setting $W_1 = W'_1 = 0$ makes it true for $m = 1$. Then, conditioned on $W_{m-1}$ and using Eq. A.1, $T^b_m \cdot S_m = T_m \cdot \mathbb{1}\{S_m = 1\}$ is stochastically dominated by $T^b_m \cdot \text{Bernoulli}(\frac{e^c}{1+e^c})$. Applying Lemma 4.9 from [30] shows that $W_m$ is stochastically dominated by $W'_m$, which proves the induction and implies the proposition's statement. □

*Proof.* In Appendix A.1.1. □

Now that we have a test to reject a claim that the generator $\mathscr{G}$ is $c$-close for the data distribution $\mathscr{D}$, we turn our attention to the second part of $\mathscr{H}$ which claims that the target model $f$ is $\varepsilon$-DP.

**Proposition 2.** *Let $\mathscr{G}$ be $c$-close, $f$ be $\varepsilon$-DP, and $T^a \triangleq A(S,X,f)$ be the guess from the membership audit. Then, for all $v \in \mathbb{R}$ and all $t$ in the support of $T$:*

$$\mathbb{P}_{S,X,T^a}\Big[\sum_{i=1}^{m} T^a_i \cdot S_i \geq v \mid T^a = t^a\Big]$$
$$\leq \mathop{\mathbb{P}}_{S' \sim Bernoulli(\frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}})^m}\Big[\sum_{i=1}^{m} t^a_i \cdot S'_i \geq v\Big] \triangleq \beta^a(m,c,\varepsilon,v,t^a)$$

*Proof.* Fix some $t^a \in \mathbb{R}^m_+$. We study the distribution of $S$ one element $i \in [m]$ at a time:

$$\begin{aligned}
&\mathbb{P}\big[S_i = 1 \mid T^a = t^a, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&= \mathbb{P}\big[T^a = t^a \mid S_i = 1, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad \frac{\mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]}{\mathbb{P}\big[T^a = t^a \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]} \\
&\leq \frac{1}{1 + e^{-\varepsilon}\frac{\mathbb{P}\left[S_i=0 \mid S_{<i}=s_{<i}, X_{\leq i}=x_{\leq i}\right]}{\mathbb{P}\left[S_i=1 \mid S_{<i}=s_{<i}, X_{\leq i}=x_{\leq i}\right]}} \\
&\leq \frac{1}{1 + e^{-\varepsilon}e^{-c}} = \frac{e^{c+\varepsilon}}{1 + e^{c+\varepsilon}}
\end{aligned}$$

The first equality uses Bayes' rule. The first inequality uses the decomposition:

$$\begin{aligned}
&\mathbb{P}\big[T^a = t^a \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] = \\
&= \mathbb{P}\big[T^a = t^a \mid S_i = 1, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad \cdot \mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad + \mathbb{P}\big[T^a = t^a \mid S_i = 0, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad \cdot \mathbb{P}\big[S_i = 0 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big],
\end{aligned}$$

and the fact that $A(s,x,f)$ is $\varepsilon$-DP w.r.t. $s$ and hence that:

$$\frac{\mathbb{P}\big[T^a = t^a \mid S_i = 0, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]}{\mathbb{P}\big[T^a = t^a \mid S_i = 1, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]} \geq e^{-\varepsilon}.$$

The second inequality uses that:

$$\begin{aligned}
&\frac{\mathbb{P}\big[S_i = 0 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]}{\mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]} \\
&= \frac{\mathbb{P}\big[X_i = x_i \mid S_i = 0, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]}{\mathbb{P}\big[X_i = x_i \mid S_i = 1, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \\
&\quad \cdot \frac{\mathbb{P}\big[S_i = 0 \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]}{\mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \\
&= \frac{\mathbb{P}\big[X_i = x_i \mid S_i = 0, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]}{\mathbb{P}\big[X_i = x_i \mid S_i = 1, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \cdot \frac{1/2}{1/2} \\
&= \frac{\mathbb{P}_{\mathscr{G}}\big[X_i\big]}{\mathbb{P}_{\mathscr{D}}\big[X_i\big]} \geq e^{-c}
\end{aligned}$$

As in Proposition 1, applying the law of total probability to introduce conditioning on $X_{\leq i}$ yields:

$$\mathbb{P}\big[S_i = 1 \mid T^a = t^a, S_{<i} = s_{<i}\big] \leq \frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}}, \tag{5.2}$$

and we can proceed by induction. Assume inductively that $W_{m-1} \triangleq \sum_{i=1}^{m-1} T_i^a \cdot S_i$ is stochastically dominated (see Definition 4.8 in [30]) by $W'_{m-1} \triangleq \sum_{i=1}^{m-1} T_i^a \cdot S'_i$, in which $S' \sim \text{Bernoulli}(\frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}})^{m-1}$. Setting $W_1 = W'_1 = 0$ makes it true for $m =$

1. Then, conditioned on $W_{m-1}$ and using Eq. A.2, $T_m^a \cdot S_m = T_m^a \cdot \mathbb{1}\{S_m = 1\}$ is stochastically dominated by $T_m^a \cdot \text{Bernoulli}(\frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}})$. Applying Lemma 4.9 from [30] shows that $W_m$ is stochastically dominated by $W'_m$, which proves the induction and implies the proposition's statement. □

*Proof.* In Appendix A.1.2. □

We are now ready to provide a test for hypothesis $\mathscr{H}$, by applying a union bound over Propositions 1 and 2:

**Corollary 1.** *Let $\mathscr{H}$ be true, $T^b \triangleq B(S,X)$, and $T^a \triangleq A(S,X,f)$. Then:*

$$\mathbb{P}\Big[\sum_{i=1}^{m} T_i^a \cdot S_i \geq v^a,\ \sum_{i=1}^{m} T_i^b \cdot S_i \geq v^b \mid T^a = t^a, T^b = t^b\Big] \\ \leq \beta^a(m,c,\varepsilon,v^a,t^a) + \beta^b(m,c,v^b,t^b)$$

To make things more concrete, let us instantiate Corollary 1 as we do in `PANORAMIA`. Our baseline ($B$ above) and MIA ($A$ above) classifiers return a membership guess in $T^{a,b} \in \{0,1\}^m$, with 1 corresponding to membership. Let us call $r^{a,b} \triangleq \sum_i t_i^{a,b}$ the total number of predictions, and $\text{tp}^{a,b} \triangleq \sum_i t_i^{a,b} \cdot s_i$ the number of correct membership guesses (true positives). We also call the precision $\text{prec}^{a,b} \triangleq \frac{\text{tp}^{a,b}}{r^{a,b}}$. Using the following tail bound on the sum of Bernoulli random variables for simplicity and clarity (we use a tighter bound in practice, but this one is easier to read),

$$\mathbb{P}_{S' \sim \text{Bernoulli}(p)^r}\Big[\sum_{i=1}^{r} \frac{S'_i}{r} \geq p + \sqrt{\frac{\log(1/\beta)}{2r}}\Big] \leq \beta,$$

we can reject $\mathscr{H}$ at confidence level $\beta$ by setting $\beta^a = \beta^b = \frac{\beta}{2}$ and if either $\text{prec}^b \geq \frac{e^c}{1+e^c} + \sqrt{\frac{\log(2/\beta)}{2r^b}}$ or $\text{prec}^a \geq \frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}} + \sqrt{\frac{\log(2/\beta)}{2r^a}}$.

### 5.1.1 Quantifying Privacy Leakage and Interpretation

Ideally in an audit we would like to quantify $\varepsilon$, not just reject a given $\varepsilon$ claim. We can use the hypothesis test from Corollary 1 to compute a confidence interval on $c$ and $\varepsilon$. To do this, we first need to define an ordering between $(c,\varepsilon)$ pairs, such that if $(c_1,\varepsilon_1) \leq (c_2,\varepsilon_2)$, the event (*i.e.*, set of observations for $T^{a,b}, S$) for which we can

reject $\mathscr{H}(c_2, \varepsilon_2)$ is included in the event for which we can reject $\mathscr{H}(c_1, \varepsilon_1)$. That is, if we can reject $\mathscr{H}$ for values $(c_2, \varepsilon_2)$ based on audit observations, we can also reject $\mathscr{H}$ for values $(c_1, \varepsilon_1)$ based on the same observations.

We define the following lexicographic order to fit this assumption, based on the hypothesis test from Corollary 1:

$$(c_1, \varepsilon_1) \leq (c_2, \varepsilon_2) \text{ if either } \begin{cases} c_1 < c_2, \text{ or} \\ c_1 = c_2 \text{ and } \varepsilon_1 \leq \varepsilon_2 \end{cases} \tag{5.3}$$

With this ordering, we have that:

**Corollary 2.** *For all $\beta \in ]0,1]$, m, and observed $t^a, t^b$, call $v^a_{ub}(c,\varepsilon) \triangleq \sup\{v : \beta^a(m,c,\varepsilon,v,t^a) \leq \frac{\beta}{2}\}$ and $v^b_{ub}(c) \triangleq \sup\{v : \beta^b(m,c,v,t^b) \leq \frac{\beta}{2}\}$. Then:*

$$\begin{aligned} &\mathbb{P}\Big[(c,\varepsilon) \geq \sup\left\{(c',\varepsilon') : t^a \cdot s \leq v^a_{ub}(c',\varepsilon') \text{ and } t^b \cdot s \leq v^b_{ub}(c')\right\}\Big] \\ &\geq 1-\beta \end{aligned}$$

*Proof.* Apply Lemma 4.7 from [30] with the ordering from Eq. 5.3 and the test from Corollary 1. □

That means that the lower bound of the confidence interval for $(c,\varepsilon)$ at confidence $1-\beta$ is the largest $(c,\varepsilon)$ pair that cannot be rejected using Corollary 1 with false rejection probability at most $\beta$. Hence for a given confidence level $1-\beta$, `PANORAMIA` computes $(c_{\text{lb}}, \tilde{\varepsilon})$, the largest value for $(c,\varepsilon)$ that it cannot reject. $(c_{\text{lb}}, \tilde{\varepsilon})$ lower-bounds the true value for $(c,\varepsilon)$ with probability at least $1-\beta$. Note that Corollaries 1 and 2 rely on a union bound between two tests, one for $c$ and one for $c+\varepsilon$. We can thus consider each test separately. In practice we follow previous practice [22, 30] for each test separately, and determine the best threshold on membership score $t^{a,b}$ considering the whole precision/recall curve. Each level of recall (threshold to predict membership based on $t^{a,b}$) corresponds to a bound on the precision, which we can compare to the empirical value. For each test separately, we pick the level of recall yielding the highest lower-bound. This is shown on lines 4-7 in the last section of Algorithm 1.

We next discuss the semantics of returned values, $(c_{\text{lb}}, \tilde{\varepsilon})$.

### 5.1.2 Audit semantics.

Corollary 2 gives us a lower-bound for $(c, \varepsilon)$, based on the ordering from Eq. 5.3. To understand the value $\tilde{\varepsilon}$ returned by PANORAMIA, we need to understand what the hypothesis test rejects. Rejecting $\mathscr{H}$ means either rejecting the claim about $c$, or the claim about $c+\varepsilon$ (which is the reason for the ordering in Eq. 5.3). With Corollary 2, we hence get both a lower-bound $c_{\text{lb}}$ on $c$, and $\{c+\varepsilon\}_{\text{lb}}$ on $c+\varepsilon$. Unfortunately, $\tilde{\varepsilon} \triangleq \{c+\varepsilon\}_{\text{lb}} - c_{\text{lb}}$, which is the value PANORAMIA returns, does not imply a lower-bound on $\varepsilon$. Instead, we can claim that "PANORAMIA could not reject a claim of $c$-closeness for $\mathscr{G}$, and if this claim is tight, then $f$ cannot be more than $\tilde{\varepsilon}$-DP".

While this is not as strong a claim as typical lower-bounds on $\varepsilon$-DP from prior privacy auditing works, we believe that this measure is useful and practical. Indeed, the $\tilde{\varepsilon}$ measured by PANORAMIA is a quantitative privacy measurement, that will be accurate (close to a lower-bound on $\varepsilon$-DP) when the baseline performs well (and hence $c_{\text{lb}}$ is tight).

When the baseline is good, we can thus interpret $\tilde{\varepsilon}$ as (close to) a lower bound on (pure) DP. In addition, since models on the same dataset and task share the same baseline, which does not depend on the audited model, PANORAMIA's measurement can be used to directly compare privacy leakage between models. Thus, PANORAMIA opens a new capability, measuring privacy leakage of a trained model $f$ without access or control of the training pipeline or the whole training set, with an interpretable and practically useful measurement.

# Chapter 6

# Experimental setup

In this chapter, we detail the target ML models we audit. We also highlight the architecture and experimental details for our generative model as well as the MIA and Baseline models that we use to quantify privacy leakage and generator data quality respectively.

## 6.1 Image classification target models

We choose target models based on the suitability of data classification task, and complexity. In addition, we also choose deeper architectures and highly overfit models along with models that generalize well to show the extent of privacy leakage measurement by PANORAMIA. We audit target models with the following architectures: a Multi-Label Convolutional Neural Network (CNN) with four layers [28], and the ResNet101 [10]. We also include in our analysis, differentially-private models for ResNet18 [10] and WideResNet-16-4 [34] models as targets, with $\varepsilon = 1, 2, 4, 6, 10, 15, 20$. The ResNet-based models are trained on CIFAR10 using $50k$ images Krizhevsky [17] of 32x32 resolution. For all CIFAR10 based classification models (apart from the DP ones), we use a training batch size of 64. The associated test accuracies and epochs are mentioned in Table 6.1. The Multi-Label CNN is trained on $200k$ images of CelebA [20] of 128x128 resolution, training batch-size 32, to predict 40 attributes associated with each image.

| ML Model | Dataset | Training Epoch | Test Accuracy | Model Variants Names* |
|---|---|---|---|---|
| ResNet101 | CIFAR10 | 20, 50, 100 | 91.61%, 90.18%, 87.93% | ResNet101-E20, ResNet101-E50, ResNet101-E100, |
| Multi-Label CNN | CelebA | 50, 100 | 81.77%, 78.12% | CNN_E50, CNN_E100 |
| MLP Tabular Classification | Adult | 10, 100 | 86%, 82% | MLP_E10, MLP_E100 |

**Table 6.1:** Train and Test Metrics for ML Models Audited. *"Model Variants" trained for different numbers of epochs $E$.

## 6.2 Generative model

For both image datasets, we use StyleGAN2 [15] to train the generative model $\mathscr{G}$ from scratch on $D_G$, and produce non-member images. For CIFAR10 dataset, we use a $10,000$ out of $50,000$ images from the training data of the target model to train the generative model. For the CelebA dataset, we select $35,000$ out of $200,000$ images from the training data of the target model to train the generative model. Generated images will in turn serve as non-members for performing the MIAs. Figure 6.1 shows examples of member and non-member images used in our experiments. In the case of CelebA, we also introduce a vanilla CNN as a classifier or filter to distinguish between fake and real images and remove any poor-quality images that the classifier detects with high confidence. The data used to train this classifier was the same data used to train StyleGAN2, which ensures that the generated high-resolution images are of high quality.

## 6.3 MIA and Baseline training

For the MIA, we follow a loss-based attack approach: `PANORAMIA` takes as input raw member and non-member data points for training along with the loss values the target model $f$ attributes to these data points. More precisely, the training set of `PANORAMIA` is:

$$(D^{tr}_{in}, f(D^{tr}_{in})) \cup (D^{tr}_{out}, f(D^{tr}_{out}))$$

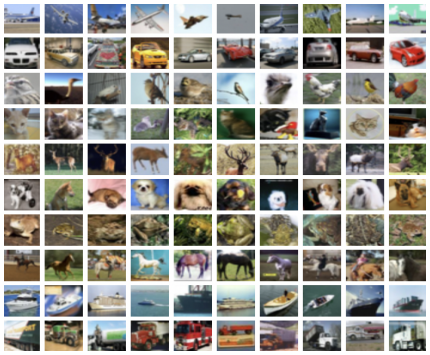
In §5.1.1, we discussed the importance of having a tight $c_{\text{lb}}$ so that our measure, $\tilde{\varepsilon}$, becomes close to a lower-bound on $\varepsilon$-DP, which requires a strong baseline. To strengthen our baseline, we introduce the helper model $h$, which helps the baseline model $b$ by supplying additional features (*i.e.*, embeddings) that can be viewed as side information about the data distribution. The motivation is that $h$'s features

might differ between samples from $\mathscr{D}$ and $\mathscr{D}'$, enhancing the performance of the baseline classifier. This embedding model $h$ is similar in design to $f$ (same task and architecture) but is trained on synthetic data that is close in distribution to the real member data distribution. Whether for the baseline or MIA, we use side information models ($h$ and $f$, respectively) by concatenating the loss of $h(x)$ and $f(x)$ to the final feature representation (more details are provided later) before a last layer of the MIA/Baseline makes the membership the prediction. Since we need labels to compute the loss, we label synthetic images with a Wide ResNet-28-2 in the case of CIFAR10, and a Multi-Label CNN of similar architecture as the target model in the case of CelebA labeling. For both instances, we used a subset of the data, that was used to train the respective generative models, to train the "labeler" classifiers as well.

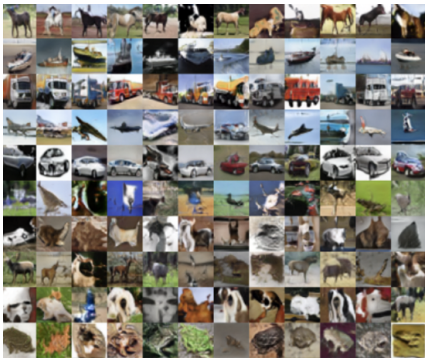
We use two different modules for both MIA and Baseline training. More precisely, the first module optimizes image classification using a built-in Pytorch ResNet101 classifier. The second module, in the form of a multi-layer perceptron, focuses on classifying member and non-member labels via loss values attributed to these data points by $f$ as input for the loss module of MIA and losses of $e$ to the baseline $b$ respectively. We then stack the scores of both image and loss modules into a logistic regression task (as a form of meta-learning) to get the final outputs for member and non-member data points by MIA and baseline $b$. The MIA and baseline are trained on 4500 data samples (half members and half generated non-members). The test dataset consists of 10000 samples, again half members and half generated non-members. The actual and final number of members and non-members that ended up in the test set depends on the Bernoulli samples in our auditing game.

(a) Real

(b) Synthetic



(c) Real



(d) Synthetic

**Figure 6.1:** Member and Non-Member datasets used in our experiments for CelebA (6.1(a), 6.1(b)) and CIFAR10 (6.1(c), 6.1(d)) image data.

# Chapter 7

# Baseline Classifier Strength Evaluation

Recall from §5.1.1 the importance of having a tight $c_{\text{lb}}$ for our measure $\tilde{\varepsilon}$ to be close to a lower-bound on $\varepsilon$-DP, which also requires a strong baseline. To increase the performance of our baseline $b$, we mimic the role of the target model $f$'s loss in the MIA using a helper model $h$, which adds a loss-based feature to $b$. This new feature can be viewed as side information about the data distribution. Table 7.1 shows the $c_{\text{lb}}$ value under different designs for $h$. The best performance is consistently when $h$ is trained on synthetic data before being used as a feature to train the $b$. Indeed, such a design reaches a $c_{\text{lb}}$ up to 1.36 larger that without any helper (CIFAR10) and 0.16 higher than when training on real non-member data *without requiring access to real non-member data*, a key requirement in `PANORAMIA`. We adopt this design in all the following experiments. In Figure 7.1 we show that the baseline has enough training data (vertical dashed line) to reach its best performance. All these pieces of evidence confirm the strength of our baseline.

| Baseline model | $c_{lb}$ |
|---|---|
| CIFAR-10 Baseline$_{D_h^{\text{tr}}=\text{gen}}$ | **2.508** |
| CIFAR-10 Baseline$_{D_h^{\text{tr}}=\text{real}}$ | 2.37 |
| CIFAR-10 Baseline$_{\text{no helper}}$ | 1.15 |
| CelebA Baseline$_{D_h^{\text{tr}}=\text{gen}}$ | **2.03** |
| CelebA Baseline$_{D_h^{\text{tr}}=\text{real}}$ | 1.67 |
| CelebA Baseline$_{\text{no helper}}$ | 0.91 |
| WikiText-2 Baseline$_{D_h^{\text{tr}}=\text{gen}}$ | **2.61** |
| WikiText-2 Baseline$_{D_h^{\text{tr}}=\text{real}}$ | 2.59 |
| WikiText-2 Baseline$_{\text{no helper}}$ | 2.34 |
| Adult Baseline$_{D_h^{\text{tr}}=\text{gen}}$ | **2.34** |
| Adult Baseline$_{D_h^{\text{tr}}=\text{real}}$ | 2.18 |
| Adult Baseline$_{\text{no helper}}$ | 2.01 |

**Table 7.1:** Baseline evaluation with different helper model scenarios

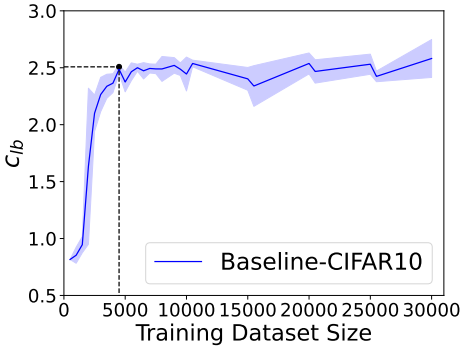



**Figure 7.1:** CIFAR-10 baseline on increasing training size.

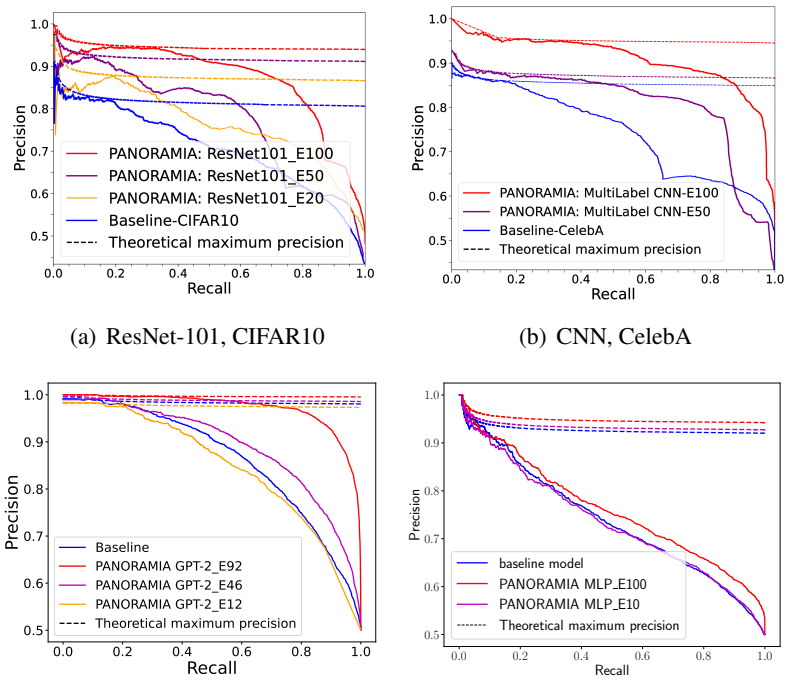
# Chapter 8

# Privacy Leakage Evaluation

We instantiate PANORAMIA on target models for four tasks from three data modalities. For *image classification*, we consider the CIFAR10 [17], and CelebA [20] datasets, with varied target models: a four-layers CNN [28], a ResNet101 [10] and a differentially-private ResNet18 [10] trained with DP-SGD [1] using Opacus [33] at different values of $\varepsilon$. We use StyleGAN2 [15] for $\mathscr{G}$. For *language models*, we fine-tune small GPT-2 [29] on the WikiText-2 train dataset [24] (we also incorporate documents from WikiText-103 to obtain a larger dataset). $\mathscr{G}$ is again based on small GPT-2, and then fine-tuned on $D_G$. We generate samples using top-$p$ sampling [11] and a held-out prompt dataset $D_G^{prompt} \subset D_G$. Finally, for *classification on tabular data*, we fit a Multi-Layer Perceptron (MLP) with 4 hidden layers trained on the Adult dataset [3], on a binary classification task predicting income $>$ \$50$k$. We use the MST algorithm [23] for $\mathscr{G}$.

Table 6.1 summarizes the tasks, models and performance, as well as the respective names we use to show results.

Our results are organized as follows. First, we evaluate the strength of our baseline, previously discussed in (§7), on which the semantics of PANORAMIA's audit rely. Second, we show what PANORAMIA detects meaningful privacy leakage in our settings, comparable to the lower-bounds provided by the O(1) approach [30] (though under weaker requirements) (§8.1). Finally, we show that PANORAMIA can detect varying amounts of leakage from models with controlled data leakage using model size and DP (§8.3).




(a) ResNet-101, CIFAR10

(b) CNN, CelebA

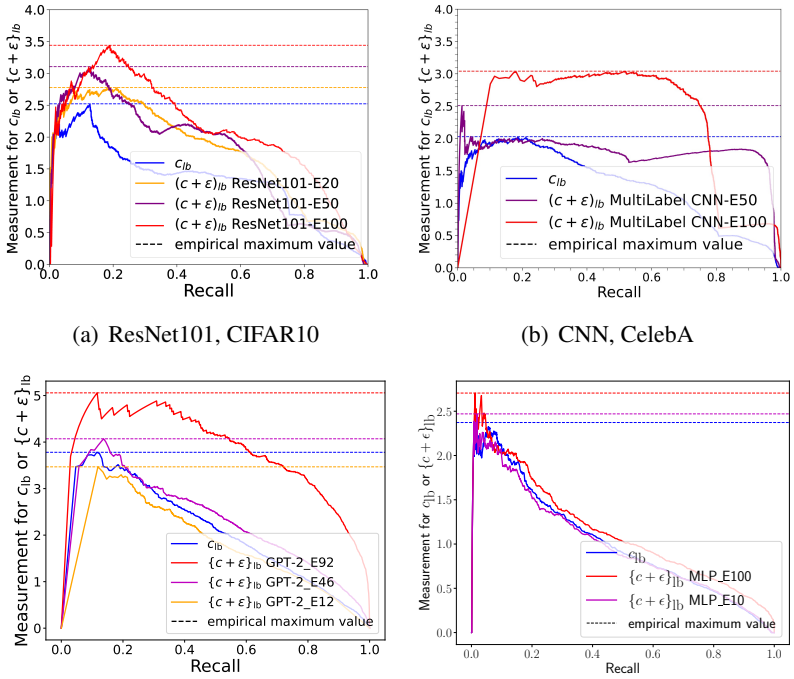
(c) GPT-2, WikiText-2

(d) MLP, Adult

**Figure 8.1:** Precision vs recall between PANORAMIA and the baseline $b$, for our target models.

## 8.1 Main Auditing Results

We run PANORAMIA on models with different values of over-fitting (by varying the number of epochs, see the final accuracy on Table 6.1) for each data modality. More over-fitted models are known to leak more information about their training data due to memorization [4, 6, 32]. To show the auditing power of PANORAMIA, we compare it with two strong approaches to lower-bounding privacy loss. First, we use a variation of our approach using real non-member data instead of generated data (called RM;RN for Real Members; Real Non-members). While this basically removes the role of the baseline ($c_{\text{lb}} = 0$), it requires access to a large sample of non-member data from the same distributions as members (hard requirement) or the possibility of training costly shadow models to create such non-members. Second, we rely on the O(1) audit from Steinke et al. [30], which is similar to the previous




(a) ResNet101, CIFAR10

(b) CNN, CelebA

(c) GPT-2, WikiText-2

(d) MLP, Adult

**Figure 8.2:** $\{c+\varepsilon\}_{lb}$ (or $c_{\text{lb}}$) vs recall, for our target models.

approach but predicts membership based on a loss threshold while also leveraging guesses on non-members in its statistical test. Note that this technique requires control of the training process. The privacy loss measured by these techniques gives a target that we hope PANORAMIA to detect.

Figure 8.1 shows the precision of $b$ and PANORAMIA at different levels of recall, and Figure 8.2 the corresponding value of $\{c+\varepsilon\}_{lb}$ (or $c_{\text{lb}}$ for $b$). Dashed lines show the maximum value of $\{c+\varepsilon\}_{lb}/c_{\text{lb}}$ achieved (Fig. 8.2) (returned by PANORAMIA), and the precision implying these values at different recalls (Fig. 8.1). Table 8.1 summarizes those $\{c+\varepsilon\}_{lb}/c_{\text{lb}}$ values, as well as the $\varepsilon$ measured by existing approaches. We make two key observations:

First, the best prior method (whether RM;RN or O(1)) measures a larger privacy loss ($\tilde{\varepsilon} \leq \varepsilon$), except on tabular data. Those surprising results are likely due to PANORAMIA's use of both raw data and the target model's loss in an ML MIA

| **Target model $f$** | Audit | $\mathbf{c}_{\text{lb}}$ | $\{\varepsilon+\mathbf{c}\}_{\text{lb}}$ | $\tilde{\varepsilon}$ | $\varepsilon$ |
|---|---|---|---|---|---|
| ResNet101_E20 | PANORAMIA | 2.508 | 2.83 | 0.32 | - |
| | PANORAMIARM;RN | 0 | 0.42 | - | 0.42 |
| | O (1) RM;RN | - | - | - | 0.52 |
| ResNet101_E50 | PANORAMIA | 2.508 | 3.15 | 0.64 | - |
| | PANORAMIARM;RN | 0 | 0.61 | - | 0.61 |
| | O (1) RM;RN | - | - | - | 0.81 |
| ResNet101_E100 | PANORAMIA | 2.508 | 3.47 | 0.962 | - |
| | PANORAMIARM;RN | 0 | 1.03 | - | 1.03 |
| | O (1) RM;RN | - | - | - | 1.40 |
| CNN_E50 | PANORAMIA | 2.01 | 2.50 | 0.49 | - |
| | PANORAMIARM;RN | 0 | | - | 0.76 |
| | O (1) RM;RN | - | - | - | 0.99 |
| CNN_E100 | PANORAMIA | 2.01 | 3.03 | 1.02 | - |
| | PANORAMIARM;RN | 0 | | - | 1.26 |
| | O (1) RM;RN | - | - | - | 1.53 |
| GPT2_E12 | PANORAMIA | 3.78 | 3.47 | 0 | - |
| | PANORAMIARM;RN | 0 | 0.30 | - | 0.30 |
| | O (1) RM;RN | - | - | - | 1.54 |
| GPT2_E46 | PANORAMIA | 3.78 | 4.07 | 0.29 | - |
| | PANORAMIARM;RN | 0 | 2.37 | - | 2.37 |
| | O (1) RM;RN | - | - | - | 4.12 |
| GPT2_E92 | PANORAMIA | 3.78 | 5.06 | 1.28 | - |
| | PANORAMIARM;RN | 0 | 3.45 | - | 3.45 |
| | O (1) RM;RN | - | - | - | 5.43 |
| MLP_E10 | PANORAMIA | 2.37 | 2.47 | 0.10 | - |
| | O (1) RM;RN | - | - | - | 0. |
| MLP_E100 | PANORAMIA | 2.37 | 2.71 | 0.34 | - |
| | O (1) RM;RN | - | - | - | 0.23 |
| MLP_E100_half | PANORAMIA | 1.25 | 1.62 | 0.37 | - |
| | PANORAMIARM;RN | 0 | 0.64 | - | 0.64 |
| | O (1) RM;RN | - | - | - | 0.22 |

**Table 8.1:** Privacy audits on different target models. Here $c_{lb}$ is the same across same datasets, where ResNet101 is trained on CIFAR10, CNN is trained on CelebA, GPT-2 on WikiText-2 and MLP on Adult Dataset. The value $\tilde{\varepsilon} = \{c+\varepsilon\}_{lb}$ - $c_{lb}$ then depends on the privacy leakage attributed to a specific target model. $\varepsilon$ is the true lower-bound when $c_{lb}$ is tight (or zero).

model, whereas O(1) uses a threshold value on the loss only. Overall, these results empirically confirm the strength of $b$, as we do not seem to spuriously assign differences between $\mathscr{G}$ and $\mathscr{D}$ to our privacy loss proxy $\tilde{\varepsilon}$. We also note that O(1) tends to perform better, due to its ability to rely on non-member detection, which improves the power of the statistical test at equal data sizes. Such tests are not available in PANORAMIA given our one-sided closeness definition for $\mathscr{G}$ (see §5), and we keep that same one-sided design for RM;RN for comparison's sake.

Second, the values of $\tilde{\varepsilon}$ measured by PANORAMIA are close to those of the methods against which we compared. In particular, despite a more restrictive

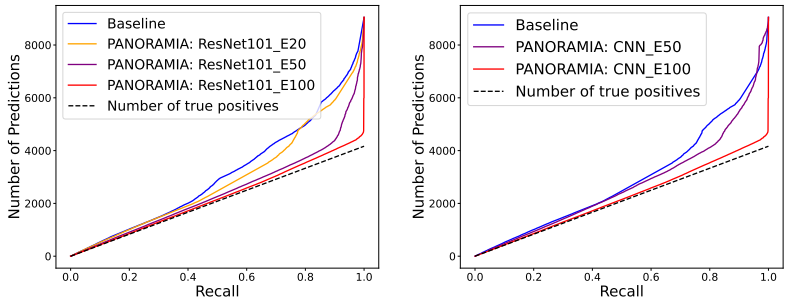
adversary model (*i.e.*, no non-member data, no control over the training process, and no shadow model training), PANORAMIA is able to detect meaningful amounts of privacy loss, comparable to that of state-of-the-art methods! For instance, on a non-overfitted CIFAR-10 model (E20), PANORAMIA detects a privacy loss of 0.32, while using real non-member (RM;RN) data yields 0.42, and controlling the training process $O(1)$ gets 0.52. The relative gap gets even closer on models that reveal more about their training data. Indeed, for the most over-fitted model (E100), $\tilde{\varepsilon} = 0.96$ is very close to RM;RN ($\varepsilon = 1.0$) and O(1) ($\varepsilon = 1.4$). This also confirms that the leakage detected by PANORAMIA on increasingly over-fitted models does augment, which is confirmed by prior state-of-the-art methods. For instance, NLP models' $\tilde{\varepsilon}$ goes from 0.18 to 1.28 (1.54 to 5.43 for O(1)), and tabular data MLPs from 0.1 to 0.34 (0 to 0.23 for O(1)).

The value of $\tilde{\varepsilon}$ for each target model is the gap between its corresponding dashed line and the baseline one in Figure 8.2. This allows us to compute values of $\tilde{\varepsilon}$ reported in Table 8.1. It is also interesting to note that the maximum value of $\{c+\varepsilon\}_{\text{lb}}$ typically occurs at low recall. Even if we do not use the same metric (precision-recall as opposed to TPR at low FPR) this is coherent with the findings from [7]. We can detect more privacy leakage (higher $\{c+\varepsilon\}_{\text{lb}}$ which leads to higher $\tilde{\varepsilon}$) when making a few confident guesses about membership rather than trying to maximize the number of members found (*i.e.*, recall). Figure 8.3, decomposes the precision in terms of the number of true positives and the number of predictions for a direct mapping to the propositions 1 and 2. These are non-negligible values of privacy leakage, even though the true value is likely much higher.

## 8.2 Privacy Auditing of Overfitted ML Models

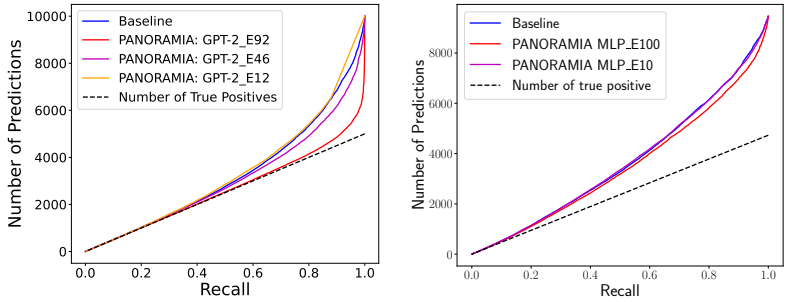
**Methodology.** Varying the number of training epochs for the target model to induce overfitting is known to be a factor in privacy loss [6, 32]. As discussed in Section 8.1, since these different variants of target models share the same dataset and task, PANORAMIA can compare them in terms of privacy leaking.

To verify if PANORAMIA will indeed attribute a higher value of $\tilde{\varepsilon}$ to more overfitted models, we train our target models for varying numbers of training epochs. The final train and test accuracies are reported in Table 6.1.




(a) Number of predictions of ResNet101 on CIFAR10

(b) Number of predictions of Multi-Label CNN on the CelebA



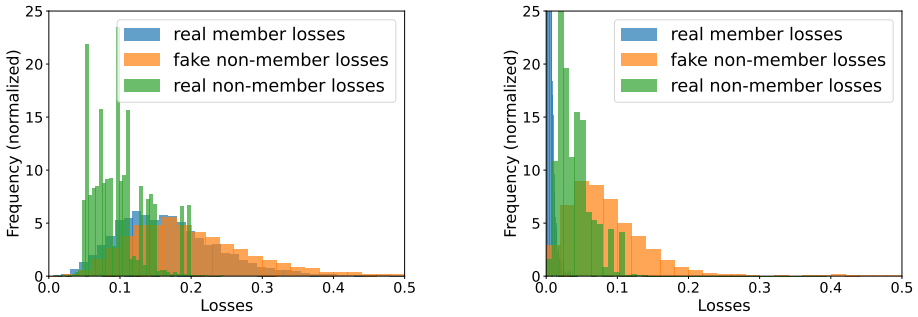

(c) Number of predictions of GPT-2 model on WikiText-2

(d) Number of predictions of an MLP on the Adult dataset.

**Figure 8.3:** Comparison of the number of true positives and predictions on different datasets

Figures 8.5 and 8.4 show how the gap between member data points (*i.e.*, data used to train the target models) and non-member data points (both real as well as generated non-members) increases as the degree of overfitting increases, in terms of loss distributions. We study the distribution of losses since these are the features extracted from the target model $f$ or helper model $h$, to pass respectively to `PANORAMIA` and the baseline classifier. The fact that the loss distributions of member data become more separable from non-member data for more overfitted models is a sign that the corresponding target model could leak more information about its training data. We thus run `PANORAMIA` on each model, hereafter presenting the results obtained.

**Results.** In Figure 8.1, we observe that more training epochs (*i.e.*, more overfit-




(a) CelebA Multi-Label CNN_E30.

(b) CelebA Multi-Label CNN_E100; an overfit model.

**Figure 8.4:** CelebA Multi-Label CNN Loss Comparisons for a generalized vs overfitted model.

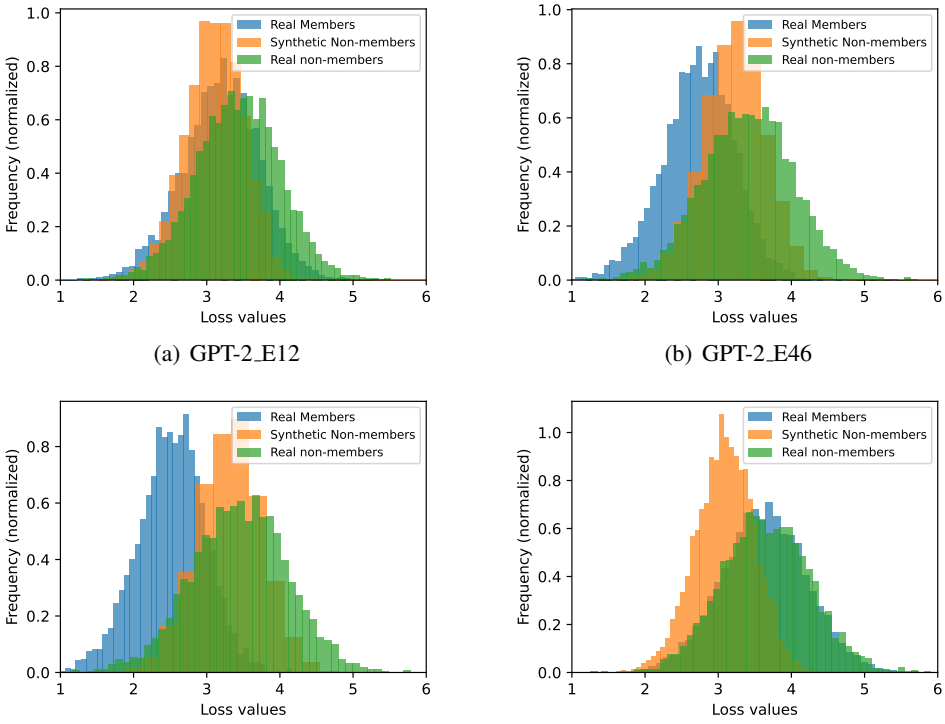
ting) lead to better precision-recall trade-offs and higher maximum precision values. Our results are further confirmed by Figure 8.3 with PANORAMIA being able to capture the number of member data points better than the baseline $b$.

In Table 8.1, we further demonstrate that our audit output $\tilde{\varepsilon}$ orders the target models in terms of privacy leakage: higher the degree of overfitting, more memorization and hence a higher $\tilde{\varepsilon}$ returned by PANORAMIA. From our experiments, we consistently found that as the number of epochs increased, the value of $\tilde{\varepsilon}$ also increased. Our experiment is coherent with the intuition that more training epochs lead to more over-fitting, leading to more privacy leakage measured with a higher value of $\tilde{\varepsilon}$.

## 8.3 Detecting Controlled Variations in Privacy Loss

**Models of varying complexity:**

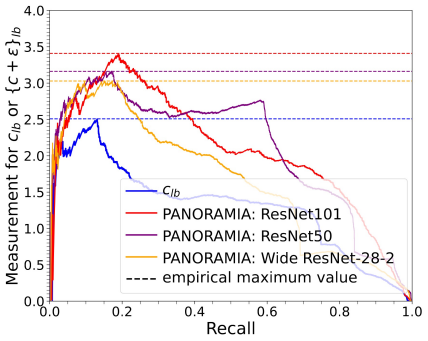
Carlini et al. [5] have shown that larger models tend to have bigger privacy losses. To confirm this, we conducted an audit of ML models with varying numbers of parameters, from a $\approx 4M$ parameters Wide ResNet-28-2, to a $25.5M$ parameters ResNet-50, and a $44.5M$ parameters ResNet-101. Figure 8.6(a) shows that PANORAMIA does detect increasing privacy leakage, with $\tilde{\varepsilon}_{wide-resnet} \leq \tilde{\varepsilon}_{resnet50} \leq \tilde{\varepsilon}_{resnet101}$.



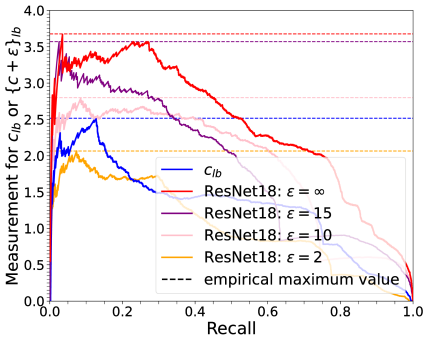

(a) GPT-2_E12 (b) GPT-2_E46

(c) GPT-2_E92 (d) Helper model $h$

**Figure 8.5:** Comparison of the loss distributions of real members, real non-members, and synthetic non-members under three target models while varying the degree of over-training on the WikiText dataset. Figure 8.5(d) compares the loss distributions under the helper model, the model providing side information to our baseline. We train the helper with some other synthetic samples, which effectively mimic real non-members' loss distributions under the target models. However, they are distinguishable to some extent from real non-members under the helper model, thus increasing our $c_{\text{lb}}$.




(a) CIFAR-10, varying model complexities.




(b) ResNet18, CIFAR-10, $\varepsilon$-DP for various $\varepsilon$ values.

**Figure 8.6:** $\{c+\varepsilon\}_{lb}$ when varying privacy leakage.

**DP models:**

Another avenue to varying privacy leakage is by training DP models with diverse values of $\varepsilon$. We evaluate PANORAMIA on DP ResNet-18 models on CIFAR10, with $\varepsilon$ values shown on Table 8.2 and Figure 8.6(b). The hyper-parameters were tuned independently for the highest train accuracy. As the results show, neither PANORAMIA nor O(1) detect privacy loss on the most private models ($\varepsilon = 1,2$). At higher values of $\varepsilon = 10,15$ (*i.e.*, less private models) and $\varepsilon = \infty$ (*i.e.*, non-private model) PANORAMIA does detect an increasing level of privacy leakage with $\tilde{\varepsilon}_{\varepsilon=10} < \tilde{\varepsilon}_{\varepsilon=15} < \tilde{\varepsilon}_{\varepsilon=\infty}$. In this regime, the O(1) approach detects a larger (except for $\varepsilon = 10$, in which PANORAMIA surprisingly detects a slightly larger value), though comparable, amount of privacy loss.

**Methodology.** We evaluate the performance of PANORAMIA on differentially-private ResNet-18 and Wide-ResNet-16-4 models on the CIFAR10 dataset under different target privacy budgets ($\varepsilon$) with $\delta = 10^{-5}$ and the non-private ($\varepsilon = \infty$) cases. The models are trained using the DP-SGD algorithm [1] using Opacus [33], which we tune for the highest train accuracy on learning rate $lr$, number of epochs $e$, batch size $bs$ and maximum $\ell_2$ clipping norm (C) for the largest final accuracy. The noise multiplier $\sigma$ is computed given $\varepsilon$, number of epochs, and batch size. Both PANORAMIA and O(1) [30] audits privacy loss with pure $\varepsilon$-DP analysis.

| Target model | Audit | $c_{lb}$ | $\varepsilon + c_{lb}$ | $\tilde{\varepsilon}$ | $\varepsilon$ |
|---|---|---|---|---|---|
| ResNet18 $\varepsilon = \infty$ | PANORAMIA RM;GN | 2.508 | 3.6698 | 1.161 | - |
| | O (1) RM;RN | - | - | - | 1.565 |
| ResNet18 $\varepsilon = 20$ | PANORAMIA RM;GN | 2.508 | 3.6331 | 1.125 | - |
| | O (1) RM;RN | - | - | - | 1.34 |
| ResNet18 $\varepsilon = 15$ | PANORAMIA RM;GN | 2.508 | 3.5707 | 1.062 | - |
| | O (1) RM;RN | - | - | - | 1.22 |
| ResNet18 $\varepsilon = 10$ | PANORAMIA RM;GN | 2.508 | 2.8 | 0.3 | - |
| | O (1) RM;RN | - | - | - | 0.14 |
| ResNet18 $\varepsilon = 6$ | PANORAMIA RM;GN | 2.508 | 1.28 | 0 | - |
| | O (1) RM;RN | - | - | - | 0.049 |
| ResNet18 $\varepsilon = 4$ | PANORAMIA RM;GN | 2.508 | 1.989 | 0 | - |
| | O (1) RM;RN | - | - | - | 0 |
| ResNet18 $\varepsilon = 2$ | PANORAMIA RM;GN | 2.508 | 2.065 | 0 | - |
| | O (1) RM;RN | - | - | - | 0.08 |
| ResNet18 $\varepsilon = 1$ | PANORAMIA RM;GN | 2.508 | 1.982 | 0 | - |
| | O (1) RM;RN | - | - | - | 0 |

**Table 8.2:** Privacy audit of ResNet18 under different values of $\varepsilon$-Differential Privacy using PANORAMIA and O(1) auditing frameworks, in which *RM* is for real member, *RN* for real non-member and *GN* for generated (synthetic) non-members.

**Results.** Tables 8.2 and 8.3 summarize the auditing results of PANORAMIA on different DP models. For ResNet-18, we observe that at $\varepsilon = 1, 2, 4, 6$ (more private models) PANORAMIA detects no privacy loss, whereas at higher values of $\varepsilon = 10, 15, 20$ (less private models) and $\varepsilon = \infty$ (a non-private model) PANORAMIA detects an increasing level of privacy loss with $\tilde{\varepsilon}_{\varepsilon=10} < \tilde{\varepsilon}_{\varepsilon=15} < \tilde{\varepsilon}_{\varepsilon=20} < \tilde{\varepsilon}_{\varepsilon=\infty}$, suggesting a higher value of $\varepsilon$ correspond to higher $\tilde{\varepsilon}$. We observe a similar pattern with Wide-ResNet-16-4, in which no privacy loss is detected at $\varepsilon = 1, 2$ and higher privacy loss is detected at $\varepsilon = 10, 15, 20, \infty$. We also compare the auditing performance of PANORAMIA with that of O(1) [30], with the conclusion drawn by these two methods being comparable. For both ResNet-18 and Wide-ResNet-16-4, O(1) reports values close to 0 (almost a random guess between members and non-members) for $\varepsilon < 10$ DP models, and higher values for $\varepsilon = 10, 15, 20, \infty$ DP models. The results suggest that PANORAMIA is potentially an effective auditing tool for DP models that has comparable performance with O(1) and can generalize to different model structures.

**Discussion.** We observe that the auditing outcome ($\tilde{\varepsilon}$ values for PANORAMIA and $\varepsilon$ for O(1)) can be different for DP models with the same $\varepsilon$ values (Table 8.4). We hypothesize that the auditing results may relate more to the level of overfitting

| Target model | Audit | $c_{lb}$ | $\varepsilon + c_{lb}$ | $\tilde{\varepsilon}$ | $\varepsilon$ |
|---|---|---|---|---|---|
| WRN-16-4 $\varepsilon = \infty$ | PANORAMIA RM;GN | 2.508 | 2.9565 | 0.448 | - |
| | O (1) RM;RN | - | - | - | 0.6408 |
| WRN-16-4 $\varepsilon = 20$ | PANORAMIA RM;GN | 2.508 | 2.95161 | 0.4436 | - |
| | O (1) RM;RN | - | - | - | 0.5961 |
| WRN-16-4 $\varepsilon = 15$ | PANORAMIA RM;GN | 2.508 | 2.91918 | 0.411 | - |
| | O (1) RM;RN | - | - | - | 0.5774 |
| WRN-16-4 $\varepsilon = 10$ | PANORAMIA RM;GN | 2.508 | 2.83277 | 0.3247 | - |
| | O (1) RM;RN | - | - | - | 0.171 |
| WRN-16-4 $\varepsilon = 2$ | PANORAMIA RM;GN | 2.508 | 2.2096 | 0 | - |
| | O (1) RM;RN | - | - | - | 0 |
| WRN-16-4 $\varepsilon = 1$ | PANORAMIA RM;GN | 2.508 | 1.15768 | 0 | - |
| | O (1) RM;RN | - | - | - | 0 |

**Table 8.3:** Privacy audit of Wide ResNet16-4 under different values of $\varepsilon$-Differential Privacy (DP) using PANORAMIA and O(1) auditing frameworks, where *RM* is for real member, *RN* for real non-member and *GN* for generated (synthetic) non-members.

than the target $\varepsilon$ values in trained DP models. The difference between train and test accuracies could be a possible indicator that has a stronger relationship with the auditing outcome. We also observe that O(1) shows a faster increase in $\varepsilon$ for DP models with higher targeted $\varepsilon$ values. We believe it depends on the actual ratio of the correct and total number of predicted samples, since O(1) considers both true positives and true negatives while PANORAMIA considers true positives only. We leave these questions for future work.

| Target model | Audit | $c_{lb}$ | $\varepsilon + c_{lb}$ | $\tilde{\varepsilon}$ | $\varepsilon$ | Train Acc | Test Acc | Diff(Train-Test Acc) |
|---|---|---|---|---|---|---|---|---|
| ResNet18-eps-20 | PANORAMIA RM;GN | 2.508 | 3.63 | 1.06 | - | 71.82 | 67.12 | 4.70 |
| | O (1) RM;RN | - | - | - | 1.22 | - | - | - |
| | PANORAMIA RM;GN | 2.508 | 2.28 | 0 | - | 71.78 | 68.08 | 3.70 |
| | O (1) RM;RN | - | - | - | 0.09 | - | - | - |
| ResNet18-eps-15 | PANORAMIA RM;GN | 2.508 | 3.63 | 1.13 | - | 69.01 | 65.7 | 3.31 |
| | O (1) RM;RN | - | - | - | 1.34 | - | - | - |
| | PANORAMIA RM;GN | 2.508 | 1.61 | 0 | | 66.68 | 69.30 | 2.62 |
| | O (1) RM;RN | - | - | - | 0.08 | - | - | - |

**Table 8.4:** DP models with the same $\varepsilon$ values can have different auditing outcomes.

## 8.4 Comparison with Privacy Auditing with One (1) Training Run: Experimental Details

We implement the black-box auditor version of O(1) approach [30]. This method assigns a membership score to a sample based on its loss value under the target model. They also subtract the sample's loss under the initial state (or generally, a randomly initialized model) of the target model, helping to distinguish members from non-members even more. In our instantiation of the O(1) approach, we only consider the loss of samples on the final state of the target model. Moreover, in their audit, they choose not to guess the membership of every sample. This abstention has an advantage over making wrong predictions as it does not increase their baseline. Roughly speaking, their baseline is the total number of correct guesses achieved by employing a randomized response $(\varepsilon, 0)$ mechanism, for those samples that O(1) auditor opts to predict. We incorporate this abstention approach in our implementation by using two thresholds, $t_+$ and $t_-$. More precisely, samples with scores below $t_+$ are predicted as members, those above $t_-$ as non-members, and the rest are abstained from prediction. We check all possible combinations of $t_+$ and $t_-$ and report the highest $\varepsilon$ among them, following a common practice [22, 35]. We also set $\delta$ to 0 and use a confidence interval of 0.05 in their test. In `PANORAMIA`, for each hypothesis test (whether for $c_{\text{lb}}$ or $\{c+\varepsilon\}_{\text{lb}}$), we stick to a 0.025 confidence interval for each one, adding up to an overall confidence level of 0.05. Furthermore, the audit set of O(1) is the same as the audit set of `PANORAMIA`.

# Chapter 9

# Conclusion

With PANORAMIA, our work introduces a brand new capability in the ML auditing toolbox: the ability to audit trained ML models (not only training algorithms) for privacy leakage of samples from a specific data distribution for a subset of the training set (*e.g.*, the data distribution of one participant in an FL model; or of a sub-population of interest), with access to only the trained model to audit, and a dataset of members of the training set from the distribution of interest. PANORAMIA does not require access to the training procedure, the full training set, or non-member data from the same distribution. It does not assume control of the training procedure (*e.g.*, to add canaries), neither does it require changing the final model or retraining new (shadow) models. PANORAMIA is thus a practical approach to quantifying privacy leakage, in settings previously not amenable to privacy audits. We also believe this new capacity can enable applications outside of direct ML privacy audits, such as the passive detection of unintended data uses, as in [18, 19].

As a first proposal for privacy audits without retraining, PANORAMIA leaves several technical questions open, that would benefit from further work and expand the capability of no retraining privacy audits. The main limitation of our approach is that the privacy loss we measure is not a proper lower-bound (see §5.1.1). Ideally, the hypothesis test would provide an upper-bound on $c$, potentially with a failure probability akin to DP's $\delta$. This seems challenging in the general case of a distribution: could we refine PANORAMIA's analysis to only focus on the audited sample, and provide an upper-bound for $c$ in this context? This would have the added

benefit of auditing privacy loss for a specific data subset, and not a distribution of a subset. It would also become meaningful to audit $(\varepsilon, \delta)$-DP. Another promising direction is to develop better generators for $c$-closeness, that also favor membership inference to strengthen the privacy audit, as well as better statistical tests. Overall, we believe that PANORAMIA introduces an important new capability for practical privacy audits, with several open directions to expand this capability.

# Bibliography


[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. → pages 1, 25, 33

[2] G. Andrew, P. Kairouz, S. Oh, A. Oprea, H. B. McMahan, and V. Suriyakumar. One-shot empirical privacy estimation for federated learning, 2023. → page 6

[3] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20. → page 25

[4] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019. → pages 5, 26

[5] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021. → page 31

[6] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership inference attacks from first principles. In *2022 2022 IEEE Symposium on Security and Privacy (SP) (SP)*, pages 1519–1519, Los Alamitos, CA, USA, may 2022. IEEE Computer Society. doi:10.1109/SP46214.2022.00090. URL https://doi.ieeecomputersociety.org/10.1109/SP46214.2022.00090. → pages 1, 26, 29

[7] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022. → page 29

[8] J. Dong, A. Roth, and W. J. Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019. → pages 1, 4

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 2006. → pages 1, 3

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. → pages 19, 25

[11] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019. → page 25

[12] M. Jagielski, J. Ullman, and A. Oprea. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33:22205–22216, 2020. → pages 1, 4, 5

[13] B. Jayaraman and D. E. Evans. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, 2019. → page 5

[14] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*. PMLR, 2015. → pages 1, 4

[15] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data, 2020. → pages 20, 25

[16] M. Kazmi, H. Lautraite, A. Akbari, M. Soroco, Q. Tang, T. Wang, S. Gambs, and M. Lécuyer. Panoramia: Privacy auditing of machine learning models without retraining, 2024. URL https://arxiv.org/abs/2402.09477. → page v

[17] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744. → pages 19, 25

[18] M. Lécuyer, G. Ducoffe, F. Lan, A. Papancea, T. Petsios, R. Spahn, A. Chaintreau, and R. Geambasu. {XRay}: Enhancing the {Web's} transparency with differential correlation. In *23rd USENIX Security Symposium (USENIX Security 14)*, 2014. → page 37

[19] M. Lecuyer, R. Spahn, Y. Spiliopolous, A. Chaintreau, R. Geambasu, and D. Hsu. Sunlight: Fine-grained targeting detection at scale with statistical confidence. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015. → page 37

[20] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. → pages 19, 25

[21] F. Lu, J. Munoz, M. Fuchs, T. LeBlond, E. Zaresky-Williams, E. Raff, F. Ferraro, and B. Testa. A general framework for auditing differentially private machine learning, 2023. → pages 1, 4, 5

[22] S. Maddock, A. Sablayrolles, and P. Stock. Canife: Crafting canaries for empirical privacy measurement in federated learning, 2023. → pages 6, 17, 36

[23] R. McKenna, G. Miklau, and D. Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021. → page 25

[24] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. → page 25

[25] M. Nasr, S. Songi, A. Thakurta, N. Papernot, and N. Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 866–882, 2021. doi:10.1109/SP40001.2021.00069. → pages 1, 4, 5

[26] M. Nasr, J. Hayes, T. Steinke, B. Balle, F. Tramèr, M. Jagielski, N. Carlini, and A. Terzis. Tight auditing of differentially private machine learning, 2023. → pages 1, 4, 6

[27] D. M. Negoescu, H. Gonzalez, S. E. A. Orjany, J. Yang, Y. Lut, R. Tandra, X. Zhang, X. Zheng, Z. Douglas, V. Nolkha, et al. Epsilon*: Privacy metric for machine learning models. *arXiv preprint arXiv:2307.11280*, 2023. → page 2

[28] K. O'Shea and R. Nash. An introduction to convolutional neural networks, 2015. → pages 19, 25

[29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. → page 25

[30] T. Steinke, M. Nasr, and M. Jagielski. Privacy auditing with one (1) training run. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. → pages 1, 4, 6, 10, 13, 14, 15, 16, 17, 25, 26, 33, 34, 36, 44, 45, 46, 47

[31] L. Wasserman and S. Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 2010. → pages 1, 4

[32] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018. → pages 26, 29

[33] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021. → pages 25, 33

[34] S. Zagoruyko and N. Komodakis. Wide residual networks, 2017. → page 19

[35] S. Zanella-Béguelin, L. Wutschitz, S. Tople, A. Salem, V. Rühle, A. Paverd, M. Naseri, B. Köpf, and D. Jones. Bayesian estimation of differential privacy, 2022. → pages 1, 4, 5, 36

# Appendix A

# Supporting Materials

## A.1 Proofs

For both Proposition 1 and Proposition 2, we state the proposition again for convenience before proving it.

### A.1.1 Proof of Proposition 1

**Proposition 1.** *Let $\mathscr{G}$ be c-close, and $T^b \triangleq B(S,X)$ be the guess from the baseline. Then, for all $v \in \mathbb{R}$ and all $t$ in the support of $T$:*

$$\mathbb{P}_{S,X,T^b}\Big[\sum_{i=1}^{m} T_i^b \cdot S_i \geq v \mid T^b = t^b\Big]$$
$$\leq \mathop{\mathbb{P}}_{S' \sim Bernoulli(\frac{e^c}{1+e^c})^m}\Big[\sum_{i=1}^{m} t_i^b \cdot S'_i \geq v\Big] \triangleq \beta^b(m,c,v,t^b)$$

*Proof.* Notice that under our baseline model $B(s,x) = \{b(x_1), b(x_2), \ldots, b(x_m)\}$, and given that the $X_i$ are i.i.d., we have that: $S_{<i} \perp\!\!\!\perp T^b_{<i} \mid X_{<i}$, since $T_i^b = B(S,X)_i$'s distribution is entirely determined by $X_i$; and $S_{\leq i} \perp\!\!\!\perp T^b_{>i} \mid X_{<i}$ since the $X_i$ are sampled independently from the past.

We study the distribution of $S$ given a fixed prediction vector $t^b$, one element

$i \in [m]$ at a time:

$$
\begin{aligned}
&\mathbb{P}\big[S_i = 1 \mid T^b = t^b, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&= \mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&= \mathbb{P}\big[X_i \mid S_i = 1, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big] \\
&\quad \frac{\mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]}{\mathbb{P}\big[X_i \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \\
&= \frac{\mathbb{P}\big[X_i \mid S_i = 1, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]\mathbb{P}\big[S_i = 1\big]}{\mathbb{P}\big[X_i \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \\
&= \frac{\mathbb{P}\big[X_i \mid S_i = 1\big]\frac{1}{2}}{\mathbb{P}\big[X_i \mid S_i = 1\big]\frac{1}{2} + \mathbb{P}\big[X_i \mid S_i = 0\big]\frac{1}{2}} \\
&= \frac{1}{1 + \frac{\mathbb{P}\left[X_i \mid S_i = 0\right]}{\mathbb{P}\left[X_i \mid S_i = 1\right]}} = \frac{1}{1 + \frac{\mathbb{P}_{\mathscr{G}}\left[X_i\right]}{\mathbb{P}_{\mathscr{D}}\left[X_i\right]}} \leq \frac{1}{1 + e^{-c}} = \frac{e^c}{1 + e^c}
\end{aligned}
$$

The first equality uses the independence remarks at the beginning of the proof, the second relies Bayes' rule, while the third and fourth that $S_i$ is sampled i.i.d from a Bernoulli with probability half, and $X_i$ i.i.d. conditioned on $S_i$. The last inequality uses Definition 3 for $c$-closeness.

Using this result and the law of total probability to introduce conditioning on $X_{\leq i}$, we get that:

$$
\begin{aligned}
&\mathbb{P}\big[S_i = 1 \mid T^b = t^b, S_{<i} = s_{<i}\big] \\
&= \sum_{x_{\leq i}} \mathbb{P}\big[S_i = 1 \mid T^b = t^b, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad \mathbb{P}\big[X_{\leq i} = x_{\leq i} \mid T^b = t^b, S_{<i} = s_{<i}\big] \\
&\leq \sum_{x_{\leq i}} \frac{e^c}{1 + e^c} \mathbb{P}\big[X_{\leq i} = x_{\leq i} \mid T^b = t^b, S_{<i} = s_{<i}\big],
\end{aligned}
$$

and hence that:

$$
\mathbb{P}\big[S_i = 1 \mid T^b = t^b, S_{<i} = s_{<i}\big] \leq \frac{e^c}{1 + e^c} \tag{A.1}
$$

We can now proceed by induction: assume inductively that $W_{m-1} \triangleq \sum_{i=1}^{m-1} T_i^b \cdot S_i$ is stochastically dominated (see Definition 4.8 in [30]) by $W'_{m-1} \triangleq \sum_{i=1}^{m-1} T_i^b \cdot S'_i$, in

which $S' \sim \text{Bernoulli}(\frac{e^c}{1+e^c})^{m-1}$. Setting $W_1 = W'_1 = 0$ makes it true for $m = 1$. Then, conditioned on $W_{m-1}$ and using Eq. A.1, $T^b_m \cdot S_m = T_m \cdot \mathbb{1}\{S_m = 1\}$ is stochastically dominated by $T^b_m \cdot \text{Bernoulli}(\frac{e^c}{1+e^c})$. Applying Lemma 4.9 from [30] shows that $W_m$ is stochastically dominated by $W'_m$, which proves the induction and implies the proposition's statement. □

### A.1.2 Proof of Proposition 2

**Proposition 2.** *Let $\mathscr{G}$ be c-close, $f$ be $\varepsilon$-DP, and $T^a \triangleq A(S,X,f)$ be the guess from the membership audit. Then, for all $v \in \mathbb{R}$ and all $t$ in the support of $T$:*

$$\begin{aligned}
&\mathbb{P}_{S,X,T^a}\Big[\sum_{i=1}^{m} T^a_i \cdot S_i \geq v \mid T^a = t^a\Big] \\
&\leq \mathop{\mathbb{P}}_{S' \sim \mathit{Bernoulli}(\frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}})^m}\Big[\sum_{i=1}^{m} t^a_i \cdot S'_i \geq v\Big] \triangleq \beta^a(m,c,\varepsilon,v,t^a)
\end{aligned}$$

*Proof.* Fix some $t^a \in \mathbb{R}^m_+$. We study the distribution of $S$ one element $i \in [m]$ at a time:

$$\begin{aligned}
&\mathbb{P}\big[S_i = 1 \mid T^a = t^a, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&= \mathbb{P}\big[T^a = t^a \mid S_i = 1, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad \frac{\mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]}{\mathbb{P}\big[T^a = t^a \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]} \\
&\leq \frac{1}{1 + e^{-\varepsilon}\frac{\mathbb{P}\left[S_i=0 \mid S_{<i}=s_{<i}, X_{\leq i}=x_{\leq i}\right]}{\mathbb{P}\left[S_i=1 \mid S_{<i}=s_{<i}, X_{\leq i}=x_{\leq i}\right]}} \\
&\leq \frac{1}{1+e^{-\varepsilon}e^{-c}} = \frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}}
\end{aligned}$$

The first equality uses Bayes' rule. The first inequality uses the decomposition:

$$\begin{aligned}
&\mathbb{P}\big[T^a = t^a \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] = \\
&= \mathbb{P}\big[T^a = t^a \mid S_i = 1, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad \cdot \mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad + \mathbb{P}\big[T^a = t^a \mid S_i = 0, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big] \\
&\quad \cdot \mathbb{P}\big[S_i = 0 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big],
\end{aligned}$$

and the fact that $A(s,x,f)$ is $\varepsilon$-DP w.r.t. $s$ and hence that:

$$\frac{\mathbb{P}\big[T^a = t^a \mid S_i = 0, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]}{\mathbb{P}\big[T^a = t^a \mid S_i = 1, S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]} \geq e^{-\varepsilon}.$$

The second inequality uses that:

$$\begin{aligned}
&\frac{\mathbb{P}\big[S_i = 0 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]}{\mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{\leq i} = x_{\leq i}\big]} \\
&= \frac{\mathbb{P}\big[X_i = x_i \mid S_i = 0, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]}{\mathbb{P}\big[X_i = x_i \mid S_i = 1, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \\
&\quad \cdot \frac{\mathbb{P}\big[S_i = 0 \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]}{\mathbb{P}\big[S_i = 1 \mid S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \\
&= \frac{\mathbb{P}\big[X_i = x_i \mid S_i = 0, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]}{\mathbb{P}\big[X_i = x_i \mid S_i = 1, S_{<i} = s_{<i}, X_{<i} = x_{<i}\big]} \cdot \frac{1/2}{1/2} \\
&= \frac{\mathbb{P}_{\mathscr{G}}\big[X_i\big]}{\mathbb{P}_{\mathscr{D}}\big[X_i\big]} \geq e^{-c}
\end{aligned}$$

As in Proposition 1, applying the law of total probability to introduce conditioning on $X_{\leq i}$ yields:

$$\mathbb{P}\big[S_i = 1 \mid T^a = t^a, S_{<i} = s_{<i}\big] \leq \frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}}, \tag{A.2}$$

and we can proceed by induction. Assume inductively that $W_{m-1} \triangleq \sum_{i=1}^{m-1} T_i^a \cdot S_i$ is stochastically dominated (see Definition 4.8 in [30]) by $W'_{m-1} \triangleq \sum_{i=1}^{m-1} T_i^a \cdot S'_i$, in which $S' \sim \text{Bernoulli}(\frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}})^{m-1}$. Setting $W_1 = W'_1 = 0$ makes it true for $m =$

1. Then, conditioned on $W_{m-1}$ and using Eq. A.2, $T_m^a \cdot S_m = T_m^a \cdot \mathbb{1}\{S_m = 1\}$ is stochastically dominated by $T_m^a \cdot \text{Bernoulli}(\frac{e^{c+\varepsilon}}{1+e^{c+\varepsilon}})$. Applying Lemma 4.9 from [30] shows that $W_m$ is stochastically dominated by $W'_m$, which proves the induction and implies the proposition's statement. □