Closed-Form Bounds for DP-SGD against Record-level Inference

Giovanni Cherubin*
Microsoft Security Response Center

Boris Köpf
Microsoft Azure Research

Andrew Paverd
Microsoft Security Response Center

Shruti Tople
Microsoft Azure Research

Lukas Wutschitz
Microsoft M365 Research

Santiago Zanella-Béguelin
Microsoft Azure Research

Abstract

Machine learning models trained with differentially-private (DP) algorithms such as DP-SGD enjoy resilience against a wide range of privacy attacks. Although it is possible to derive bounds for some attacks based solely on an (ε, δ)-DP guarantee, meaningful bounds require a small enough privacy budget (i.e., injecting a large amount of noise), which results in a large loss in utility. This paper presents a new approach to evaluate the privacy of machine learning models against specific record-level threats, such as membership and attribute inference, without the indirection through DP. We focus on the popular DP-SGD algorithm, and derive simple closed-form bounds. Our proofs model DP-SGD as an information theoretic channel whose inputs are the secrets that an attacker wants to infer (e.g., membership of a data record) and whose outputs are the intermediate model parameters produced by iterative optimization. We obtain bounds for membership inference that match state-of-the-art techniques, whilst being orders of magnitude faster to compute. Additionally, we present a novel data-dependent bound against attribute inference. Our results provide a direct, interpretable, and practical way to evaluate the privacy of trained models against specific inference threats without sacrificing utility.

1 Introduction

Privacy of training data is a central concern when deploying Machine Learning (ML) models. Privacy risks encompass a variety of adversary goals with corresponding threat models. For example, if one wanted to prevent an attacker with access to a model from inferring whether a specific data record was in the training data, we would aim to train the model to make it resilient against *membership inference* attacks [26, 29, 34]. On the other hand, if the concern is an attacker uncovering sensitive attributes about training data records, we would ensure resilience against *attribute inference* attacks [16, 34].

*Corresponding author.

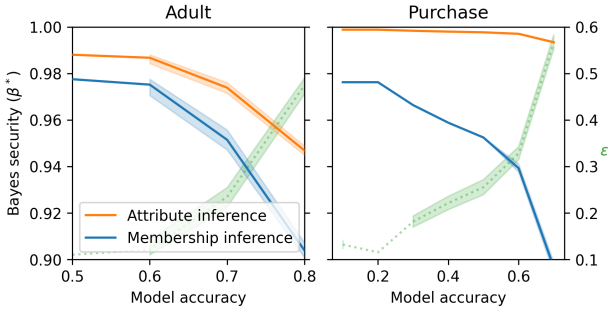

Figure 1: Bayes Security (β^*) of DP-SGD against MIA and AI on the `Adult` and `Purchase` datasets, w.r.t. the accuracy of the model; a higher β^* means a more secure model. When possible, picking the weaker AI threat model enables achieving a better privacy-utility trade-off. For reference, we report the corresponding (ε, δ)-DP (dashed green line), for $\delta = 3.8 \times 10^{-6}$ (`Adult`) and $\delta = 4 \times 10^{-7}$ (`Purchase`).

In practice one may be mostly concerned about some *specific* privacy risks, such as membership or attribute inference. However, because practitioners lack tools to analyze and mitigate these specific risks, they resort to enforce Differential Privacy (DP), which regards *any* leakage of information about individual records as a privacy violation. From a theoretical perspective, this choice is convenient: with suitable parameters (ε, δ), DP provides quantifiable resilience against all threats to individual training data records. There are numerous ways of numerically accounting for the privacy budget (ε, δ) spent when training a model [14, 17, 25], but few ways of computing bounds against specific privacy attacks [27].

Threat-agnostic. Firstly, the definition of (ε, δ)-DP is generally applied in a threat-agnostic manner. However, in practice there are cases where specific threats give rise to privacy concerns while others do not. For example, the fact that a person participated in the Census dataset is not privacy sensitive; but if an attacker were able to infer the values of sensitive

attributes such as race or age, we would rightly regard this as a privacy violation. Furthermore, there is no principled way to choose *interpretable* values for (ε, δ) without considering a specific privacy threat. Even when the threat is specified, we still need to find a relationship between this threat and (ε, δ) in order to evaluate the risk; for example, prior work has explored the relationship between DP and membership inference [6, 20, 36]. This raises the question: if our aim is to protect against specific threats, can we evaluate our models directly against these threats?

Implementation challenges. Secondly, it is known that implementations of (ε, δ)-DP accountants can be error-prone. This may be due to implementation difficulties [15, 24] or numerical errors (e.g., floating point precision) [17]. Further, despite being considered optimal (up to discretization error), accountants generally come with computational costs, which researchers are currently trying to reduce [15].

Our approach. In this paper, we show that it is possible to directly evaluate a trained model against specific privacy threats, such as membership and attribute inference, without actually performing these (often computationally expensive) attacks. We focus on the mainstream training algorithm DP-SGD, and derive simple closed-form bounds against these threats. At the core of our proof technique is the approximation of the distribution of intermediate gradients produced by DP-SGD with a Gaussian distribution. We characterize the approximation error, and show that it can be made negligible by tuning the privacy parameters of the algorithm; importantly, the error gets smaller for parameters that ensure good privacy.

Our theoretical analyses are facilitated by use of the *Bayes security* metric (β^*) [6]. The main benefit of this metric is its interpretability: it corresponds to the complement of the attacker's advantage, which is widely used in the privacy-preserving ML literature (e.g., [34]). Furthermore, Bayes security is threat model specific, prior independent, and one can easily match it to the (optimal) attacker's accuracy for a specific prior. Additionally, we prove that Bayes security bounds the true positive rate (TPR) of an attacker aiming for a certain false positive rate (i.e. TPR@FPR), which captures particularly well the risk of membership inference [5].

Overall, the simplicity of our proofs suggests our techniques can be extended to study other algorithms and privacy threats. We summarize our contributions as follows:

- We propose a new approach to directly measure the privacy of ML models trained using DP-SGD, which addresses the drawbacks outlined above: 1) It is *threat-specific*. 2) It streamlines the proof, in that the metric is directly computed without going via (ε, δ), and it makes for a straightforward implementation. Importantly, it is orders of magnitude faster to compute than state-of-the-art methods for measuring the risk against MIA [14, 17].
- We demonstrate that our new approach matches (tight) existing techniques in computing bounds for membership inference (MIA) while requiring orders of magnitude lesser computation time than prior work.
- We show a relationship between Bayes security and TPR@FPR, a standard metric for MIA [5].
- We use our new approach to compute bounds for attribute inference (AI). From our bounds, we observe that DP-SGD is significantly more secure against AI than MIA. This is important because, if a practical application requires security against AI but not MIA, one can achieve a better utility whilst maintaining acceptable privacy, as shown in Figure 1.

Our results, as well as those in the previous literature, assume that an *attacker* has access to intermediate model weights during training. However, a more realistic (*inference-time*) attacker only has access to the final weights of the model. To assist future research effort, we also report on our unsuccessful attempts towards obtaining tighter bounds for inference-time attackers. We show how our framework can model this scenario, and discuss what problems one may need to solve in order to obtain such tighter bounds.

2 Background and Preliminaries

We study the security of DP-SGD against record-level inference, with a focus on MIA and AI. In this section, we provide an overview of the DP-SGD algorithm, we formally define MIA and AI, and we describe the security metric we use to quantify the resilience of an ML model against both threats.

2.1 DP-SGD

Proposed by Abadi et al. [1], Differentially Private Stochastic Gradient Descent (DP-SGD) is a modification of SGD to satisfy (ε, δ)-DP, as shown in Algorithm 1. Consider a training set of data records $\{z_1, \ldots, z_N\} \in \mathcal{Z}^N$ and a loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, z_i)$, based on model weights θ. Let η_t be a learning rate, σ a noise scaling factor, C a gradient clipping norm, and L/N a sampling factor. DP-SGD trains a model θ_T as follows: for each step $t = 1, ..., T$, sample on average L records from the training set, clip their gradients' norms to C, and add Gaussian noise to their sum; use the resulting *noisy gradient* $\tilde{\mathbf{g}}_t$ to update the model weights according to the learning rate, and repeat for the desired number of steps.

Typically, the privacy parameters (ε, δ) are obtained numerically via *accounting mechanisms*; due to the iterative nature of DP-SGD this is essential to obtain accurate privacy guarantees. Abadi et al. [1] introduced the Moments Accountant for computing the privacy guarantees for composed mechanisms. More recently, Dong et al. [13] introduced f-DP which gives rise to lossless composition: this notion of DP composes all

Algorithm 1: DP-SGD($\{z_1,\ldots,z_N\}$, $\mathcal{L}(\theta)$, η_t, σ, L, C)

Initialize θ_0 randomly
for $t \in [T]$ **do**
  Take a random sample L_t with sampling probability L/N
  Compute gradient
  For each $i \in L_t$, compute $\mathbf{g}_t(z_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, z_i)$
  Clip gradient
  $\bar{\mathbf{g}}_t(z_i) \leftarrow \mathbf{g}_t(z_i) / \max\left(1, \frac{\|\mathbf{g}_t(z_i)\|_2}{C}\right)$
  Add noise
  $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(z_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
  Descent
  $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
end
Output θ_T

possible (ε, δ) at once, and only afterwards it converts the privacy guarantee back to a single (ε, δ) pair. Alas, computing this composition is challenging, and several numerical approximations have been developed [14, 17, 24, 25]; these are tight up to discretization error.

2.2 Threat Models

We consider two specific threat models: membership inference (Game 2), and attribute inference (Game 3).

Membership inference (MIA). In (record-level) MIA, the attacker aims to ascertain whether a data record appeared in the model's training set. This threat model is formalized in Game 2. In this game, a challenge point z_s^* is sampled from a set of challenge points $\{z_i^*\}_{i=1}^M$ according to an arbitrary prior distribution π on this set. The model is trained on $D \cup \{z_s^*\}$ for T steps, and the intermediate DP-SGD updates $\{\theta_t\}_{t=1}^T$ are revealed to the attacker; the attacker is also assumed to know the game parameters $\mathcal{T}, T, \pi$, as well as the set of challenge points. The attacker's goal is to guess *which* of the challenge points was used for training the model. Game 2 generalizes common MIA setups in two ways. First, the number of challenge points M can be larger than 2. Second, the game enables associating a prior distribution π to the choice of the challenge points. Thanks to the metric we use (Section 2.3), it will suffice to compute the security against MIA for the two worst-case challenge points ($M = 2$) and a uniform prior (Section 3).

Game 2: MIA-record-level($\mathcal{T}_T, D, \{z_1^*,\ldots,z_M^*\}, \pi_{\{1,\ldots,M\}}$)

$s \leftarrow \pi_{\{1,\ldots,M\}}$
$\{\theta_t\}_{t=1}^T \leftarrow \mathcal{T}_T(D \cup \{z_s^*\})$
$s' \leftarrow \text{Attacker}(\{\theta_t\}_{t=1}^T,\ \{z_i^*\}_{i=1}^M, D, \mathcal{T}_T, \pi_{\{1,\ldots,M\}})$

Attribute inference (AI). Let $z^* = \varphi \mid s$ be a data record, composed of the concatenation of two vectors: φ and s. In attribute inference, the attacker aims to infer the value of one or more *sensitive attributes* of a data record, s, given access to the remainder of that record, φ. As shown in Game 3, the sensitive attribute $s \in \mathcal{A}$ is sampled according to some prior π on the set. The model is trained for T steps on the training set $D \cup \{z^*\}$, and the intermediate updates $\{\theta_t\}_{t=1}^T$ are revealed to the attacker; the attacker is also assumed to know all the game parameters, including the set of sensitive attributes. The goal of the attacker is to guess the sensitive attribute s.

Game 3: AI($\mathcal{T}_T, D, \varphi, \mathcal{A}, \pi_{\mathcal{A}}$)

$s \leftarrow \pi_{\mathcal{A}}$
$z^* \leftarrow \varphi \mid s$ `// The attributes are concatenated`
$\{\theta_t\}_{t=1}^T \leftarrow \mathcal{T}_T(D \cup \{z^*\})$
$s' \leftarrow \text{Attacker}(\{\theta_t\}_{t=1}^T,\ \varphi, \mathcal{A}, D, \mathcal{T}_T, \pi_{\{1,\ldots,M\}})$

2.3 The Bayes security metric

We define a metric of risk for these threats.

Generalized attacker advantage. The commonly-used metric of *advantage* quantifies how much more likely an attacker is to succeed, at either membership or attribute inference, when given access to the trained model, as compared with not having this access. Formally, suppose the attacker's goal is to guess some secret information, measured by random variable S. Let π denote any prior knowledge the attacker has about S; mathematically, π is a probability distribution on the range of S. We write Attacker(π, θ) to indicate an attacker who has access to the model θ (and with prior knowledge π), and Attacker(π) for an attacker who guesses purely based on prior knowledge; we assume the former is at least as successful as the latter. The *generalized* advantage [7] is defined to be the difference between the probability of success of these two attackers, normalized to a value between $[0, 1]$: 0 implies no advantage and 1 maximal advantage:

$$\text{Adv}_\pi = \frac{Pr[\text{Attacker}(\pi,\theta) = S] - Pr[\text{Attacker}(\pi) = S]}{1 - Pr[\text{Attacker}(\pi) = S]}.$$

This is a generalized version of the notion of advantage typically used in the literature (e.g., [34]): by letting S take a binary value and setting π to be a uniform distribution over the possible values of S, we get $Pr[\text{Attacker}(\pi) = S] = 1/2$; substituted into the above, this gives the familiar expression for advantage, as used by Yeom et al. [34]:

$$\text{Adv}_\pi = 2Pr[\text{Attacker}(\pi,\theta) = S] - 1$$

Note that this specific notion of advantage cannot be applied to Games 2 and 3 because, in both cases, the secret may

Table 1: Summary of notation.

Symbol	Meaning
z^*	Challenge point about which the attacker wishes to learn some property.
$f(z^*)$	Property of interest.
$O_1, ..., O_T$	Intermediate weights output by DP-SGD.
$G_1, ..., G_T$	Intermediate (noisy) gradients output by DP-SGD for the challenge point.
$P_{G(z^*)\|S=s}$	Distribution of the gradient vector $G = (G_1, ..., G_T)$ given a point z^* s.t. $f(z^*) = s$.
σ	Noise parameter.
C	Gradient norm clipping parameter.
$p = L/N$	Sampling rate; N: training set size, L: user-chosen parameter.

have more than two possible values, and they need not come from a uniform prior distribution π.

Bayes Security. Based on the notion of generalized advantage, we use the *Bayes security* metric [6, 7], defined as:

$$\beta^* = 1 - \max_{\pi} \mathrm{Adv}_{\pi} \tag{1}$$

This metric takes values in the range $[0, 1]$, where 1 indicates perfect security (i.e., no information leakage). Importantly, the following holds:

Theorem 1 (Bayes security and advantage [6, Theorem 1])**.** *Bayes security is achieved on (equivalently, the generalized advantage is maximized on) a uniform prior on two secrets:*

$$\beta^* = 1 - \max_{s_0, s_1 \in \mathbb{S}} \mathrm{Adv}_{u_{s_0,s_1}},$$

where u_{s_0,s_1} is a uniform prior on two secrets $s_0, s_1 \in \mathbb{S}$, i.e., $Pr[S = s_0] = Pr[S = s_1] = 1/2$, and $Pr[S = s] = 0 \; \forall s \neq s_0, s_1$.

Conveniently, by using Bayes security we further inherit the following relations: 1) Bayes security is directly related to the total variation distance between the two worst-case distributions in the outputs of DP-SGD (Section 3), and 2) it is related to (ε, δ)-DP (Section 4).

3 Proof Strategy and Main Result

In this paper, we derive bounds on the security of DP-SGD against record-level MIA (Game 2) and AI (Game 3). First, observe that these threats can be unified in a record-level property inference setup as follows. Let z^* be some data record (i.e., the *challenge point*) about which the attacker aims to infer some property $f(z^*)$. In MIA, the challenge point is chosen from a set of possible challenge points $z^* \in \{z_1^*, ..., z_M^*\}$, and $f(z^*)$ is an index to that set such that $z^* = z^*_{f(z^*)}$. In AI, for an arbitrary challenge point z^*, composed of the concatenation $z^* = \varphi \mid s$, where $s \in \mathcal{A}$ represents a sensitive attribute, the property is $f(z^*) = s$. In our main result, we assume that f is a bijection: there is exactly one challenge point $f(z^*)$ for each property $s \in \mathrm{codom}(f)$; observe that this is satisfied by construction in our two threat models.[1]

Let $S = f(z^*)$ be a random variable representing the secret property. The attacker aims to guess S given the intermediate models output by DP-SGD, which we denote by the random vector $O = (O_0, O_1, ..., O_T)$. In the spirit of quantitative information flow [30], this can be seen as an information theoretic channel, where the relation between S and O is ruled by the posterior distribution $P_{O|S}$. The Bayes security of this channel, $\beta^*(P_{O|S})$, measures the *additional leakage* about the secret S that an attacker can exploit by observing O.

Looking ahead. In this section, we prove our main result: a bound on the Bayes security of DP-SGD against the record-level property inference attack; we later specialize this bound to the cases of MIA (Corollary 7) and AI (Corollary 10). We proceed as follows:

1. First, we show one only needs to measure the risk for the two worst-case property values (challenge points) (Section 3.1), as a consequence of Theorem 1.

2. We show that the noise on the model weights coming from the training set can be neglected for our analysis: it suffices to compute Bayes security for the gradient of the challenge point z^*. We then observe that gradients follow a Gaussian mixture distribution.

3. We prove that a mixture of Gaussians can be approximated by a single Gaussian distribution, with an error term that gets smaller as the noise parameter (σ) increases (Proposition 4).

4. We obtain a bound on the Bayes security of DP-SGD by bounding the total variation distance between the distribution of the gradients for the two challenge points that correspond to the two worst-case property values.

3.1 All You Need Is Two Points ...

In both Games 2 and 3, S takes values from a potentially large set. Further, its prior distribution may be skewed: some values of S may be more likely than others. For example, in MIA, one data record may be *more likely* than another to be a member; this is captured by our formalization in Game 2, where π may assign more weight to one particular record.

[1] We suspect that this assumption can be relaxed: if f was not a bijection, there may be two challenge points $z_0^* \neq z_1^*$ satisfying the same property $f(z_0^*) = f(z_1^*) = s$; but this can only make the attack easier, as it skews the prior distribution on s, thereby making it a more probable guess.

To solve this issue, we apply the fact that the generalized advantage is maximized over a uniform prior; this is an immediate consequence of Theorem 1 by Chatzikokolakis et al. [6]. This implies that, when studying the security of DP-SGD against these threats, it is sufficient to limit the range of S to the two values that are the easiest to distinguish for the attacker, and set π to the uniform prior.

For example, under the MIA threat model, this means that by measuring the security for just two challenge points ($M = 2$) that are equally likely to be members, we obtain a bound on the security for arbitrary values of $M \geq 2$. Equivalently, for AI, it is sufficient to look at the two *leakiest* attribute values.

3.2 ... and Intermediate Gradients

The second step in our analysis is the observation that an attacker obtains maximal advantage if they are given direct access to the intermediate gradients, rather than model weights. Formally, the attacker observes a random vector $O(z^*) = (O_0(z^*), ..., O_T(z^*))$, the intermediate model weights; in our notation, we make the dependence on the challenge point z^* explicit where needed. O is such that $O_0 = \theta_0$ and for $t \geq 0$

$$O_t(z^*) = \frac{1}{L}\left(\sum_{i=1}^{N-1} \bar{\mathbf{g}}_t(z_i)\mathcal{B}(p) + \bar{\mathbf{g}}_t(z^*)\mathcal{B}(p) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$$

where $\mathcal{B}(p)$ a Bernoulli distribution, with $p = \frac{L}{N}$, and $\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$ is isotropic Gaussian noise with variance $\sigma^2 C^2$.

We use the notation $P_{O(z^*)|S=s}$ to indicate the distribution of the intermediate weights, conditioned on the fact that the challenge point satisfies $z^* : f(z^*) = s$.

Now, consider the random vector $G = (G_1, ..., G_T)$:

$$G_t = \bar{\mathbf{g}}_t(z^*)\mathcal{B}\left(\frac{L}{N}\right) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \tag{2}$$

We observe that the distribution $P_{O|S}$ can be obtained via postprocessing from $P_{G|S}$; since the Bayes security of a channel cannot decrease by postprocessing, we have:

Corollary 2 (Consequence of Theorem 4 in [6])**.**

$$\beta^*(P_{O|S}) \geq \beta^*(P_{G|S}).$$

Intuitively, an attacker has a better (or equal) advantage when attacking channel $P_{G|S}$ than $P_{O|S}$. The reason is that G carries at least as much information about the challenge point z^* as O. We shall henceforth study the security of $P_{G|S}$.

Gradients distribution. We provide an explicit expression for $P_{G(z^*)}$, for a generic challenge point z^*.

At each step, the intermediate gradient is a Gaussian, centered either in $\bar{\mathbf{g}}(z^*)$ with probability p, or 0 otherwise. This means that P_G is a mixture of 2^T Gaussians: intuitively, G takes values from a Gaussian centered in $(0, ..., 0)$ with probability $(1-p)^T$ (i.e., the gradient is never sampled), from a Gaussian centered in $(\bar{\mathbf{g}}(z^*), 0, ..., 0)$ with probability $p(1-p)^{T-1}$ (i.e., $\bar{\mathbf{g}}(z^*)$ is sampled in the first step only), and so on. Let $b \in \{0,1\}^T$ be a binary vector, where $b_t = 1$ means that G_t is centered in $\bar{\mathbf{g}}(z^*)$, and $b_t = 0$ means that G_t is centered in 0; here, the role of b is that of a mask that indicates in which steps the challenge point is sampled. Then we can write the distribution as:

$$P_G = \sum_{b \in \{0,1\}^T} c_b \prod_{t=1}^{T} \mathcal{N}(\bar{\mathbf{g}}(z^*) \odot b, \sigma^2 I),$$

where $x \odot y$ is the Hadamard (i.e., entrywise) product of vectors x and y, and c_b is the probability of observing b, $c_b = p^{|b|}(1-p)^{T-|b|}$, where $|b|$ is the number of 1's in b.

3.3 Bayes Security of DP-SGD

To determine the Bayes security of DP-SGD, we will use the following relation.

Proposition 3 (Bayes security and total variation [6])**.** *Let $\mathcal{M} : \mathbb{S} \to \mathbb{O}$ be a randomized algorithm. Then:*

$$\beta^*(\mathcal{M}) = 1 - \max_{s_0, s_1 \in \mathbb{S}} \mathrm{tv}(P_{\mathcal{M}(S)|S=s_0}, P_{\mathcal{M}(S)|S=s_1}),$$

where $P_{\mathcal{M}(S)|S}$ is the posterior distribution of the mechanism's output $\mathcal{M}(S)$ given some input random variable S, and tv *is the total variation distance*[2].

Based on this, we compute the Bayes security of DP-SGD as the maximal total variation between $P_{G|S=s_0}$ and $P_{G|S=s_1}$, across all pairs $s_0, s_1 \in \mathbb{S}$; as observed above, $P_{G|S}$ is a mixture of Gaussians. Unfortunately, there are no known tight bounds on the divergence between mixtures of Gaussians.

This is not an unknown obstacle: all previous DP-SGD analyses (e.g., DP-based) have encountered its analog. FFT-based accountant methods address this issue by discretization: for a fine enough grid, one can empirically measure the divergence between the distributions. Recent work by Mahloujifar et al. [27] uses Monte Carlo estimations, by sampling from the mixture distribution. Both approaches, although valid, come with high computational costs.

In this paper, we study the benefits of a different strategy: we observe that the mixture distribution generated by DP-SGD can be approximated with a Gaussian distribution *for certain choices of parameters*. Fortunately, these parameter choices happen to be of interest for most practical purposes.

Approximating a mixture with a Gaussian. The proof of our main result relies on computing the total variation between two Gaussian mixtures. Our first observation is that, in

[2]Consider two measures P and Q on the same measurable space $(\mathcal{Z}, \mathcal{F})$; their total variation distance is: $\mathrm{tv}(P, Q) = \sup_{A \in \mathcal{F}} |P(A) - Q(A)|$.

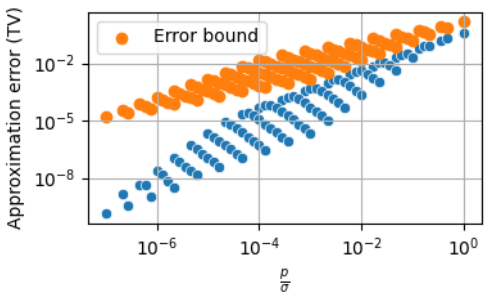

Figure 2: We compare the error induced by approximating a mixture of Gaussians with a Gaussian (Proposition 4). The error, measured as the total variation between the original and approximate distributions, is computed via numerical integration for a fixed $T = 1$. A small ratio between the sampling rate p and the noise parameter σ ensure the error is negligible.

some cases, a mixture of Gaussians can be approximated by a Gaussian. We formalize this in the following result, which shows the error committed when making this approximation in terms of the total variation between the original and approximate distributions. For clarity, we let $p = L/N$.

Proposition 4. *Let $f_{\mathcal{M}}$ be a Gaussian mixture defined as follows. For a mean vector $\mu = (\mu_1, \ldots, \mu_T)$ and covariance matrix $\sigma^2 C^2 \mathbf{I}_T$, and $C = \max_{j=1}^{T} \mu_j$, let $f_{\mathcal{M}}(x) = \sum_{b \in \{0,1\}^T} \pi_b f_{\mathcal{N}(\mu b, \sigma^2 C^2)}(x)$. The i-th component takes values from $f_{\mathcal{N}(\mu_i, \sigma^2 C^2)}$ with probability $p \in [0,1]$, or from $f_{\mathcal{N}(0, \sigma^2 C^2)}$ otherwise. Here, $\pi_b = p^{|b|}(1-p)^{T-|b|}$. The error committed in approximating $f_{\mathcal{M}}$ with $f_{\mathcal{N}(p\mu, \sigma^2 C^2)}$ is:*

$$\mathrm{tv}(f_{\mathcal{M}}, f_{\mathcal{N}(p\mu, \sigma^2 C^2)}) = O\left(\frac{\sqrt{pT}}{\sigma}\right)$$

Proofs are in the appendix.

The total variation between two Gaussians. The Bayes security of DP-SGD reduces to computing the total variation between two Gaussian distributions that are identically-scaled (with isotropic covariance matrix). For this step, we use the following closed form expression, which was derived by Devroye et al. [12] using a result by Barsov and Ul'yanov [4]:

Corollary 5 (From Barsov and Ul'yanov [4], Theorem 1)**.** *Let $\mu_0, \mu_1 \in \mathbb{R}^{m \times n}$, $\sigma > 0$. Then,*

$$\mathrm{tv}\left(\mathcal{N}(\mu_0, \sigma^2 \mathbf{I}), \mathcal{N}(\mu_1, \sigma^2 \mathbf{I})\right) = \mathrm{erf}\left(\frac{\|\mu_0 - \mu_1\|_F}{2\sqrt{2}\sigma}\right)$$

where $\|A\|_F = \sqrt{\mathrm{tr}(AA^\intercal)}$ is the Frobenius norm.

Main result. We can now state our main result: a closed-form bound on the security of DP-SGD against record-level property inference. The bound depends on a variable, Δ_f, whose value depends on the threat model (and, consequently, property of interest f), defined as:

$$\Delta_f = \max_{s_0, s_1 \in \mathrm{codom}(f)} \|\bar{\mathbf{g}}(f^{-1}(s_0)) - \bar{\mathbf{g}}(f^{-1}(s_1))\|_F \,. \quad (3)$$

Here, $\bar{\mathbf{g}}(z^*) = (\bar{\mathbf{g}}_1(z^*), \ldots, \bar{\mathbf{g}}_T(z^*))$ is the sequence of gradients computed by DP-SGD on a challenge point z^*; $f^{-1}(s)$ is the challenge point that satisfies $f(f^{-1}(s)) = s$, which is unique by assumption.

Intuitively, Δ_f indicates how much influence each property value has on the gradients, and it takes higher values the more the gradients change when the property value changes; in particular, it captures the worst-case scenario, when the attacker has to distinguish between the two property values that leak the most information. We will provide an explicit value for Δ_f for the case of MIA (Corollary 7) and AI (Corollary 10) in the next sections.

The Bayes security of DP-SGD against record-level property inference is as follows:

Theorem 6. *Assume that f is a bijection, and let Δ_f be defined as in Equation (3). The Bayes security of DP-SGD with respect to the record-level property inference threat described in Section 3 is:*

$$\beta^*(P_{O|S}) \geq 1 - \mathrm{erf}\left(p\frac{\Delta_f}{2\sqrt{2}\sigma C}\right) - O\left(\frac{\sqrt{pT}}{\sigma}\right).$$

The proof combines the approximation of a mixture with a Gaussian (Proposition 4) with the bound on the total variation between two Gaussians (Corollary 5).

In the next two sections, we apply this result to bound the security against MIA (Section 4) and AI (Section 5).

4 Membership Inference

Theorem 6 gives a bound for the Bayes security of DP-SGD against a record-level property inference attack. In this section, we apply this result to derive a bound on MIA, we study its tightness in comparison with existing estimates based on DP accountants, and we show how our bound relates to other metrics, such as the TPR@FPR of an optimal attacker.

The Bayes security of DP-SGD against MIA follows as a corollary to Theorem 6:

Corollary 7. *The Bayes security of DP-SGD against record-level MIA (Game 2) is:*

$$\beta^* \geq 1 - \mathrm{erf}\left(p\frac{\sqrt{T}}{\sqrt{2}\sigma}\right) - O\left(\frac{\sqrt{pT}}{\sigma}\right).$$

DP-SGD parameters selection. Theorem 6 makes it possible to *cheaply* decide on which parameters to select before running DP-SGD, given a desired level of MIA-resilience.

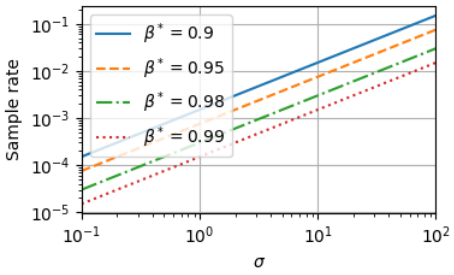

Figure 3: Bayes security against MIA: picking the noise and sampling rate to achieve a desired level of security ($T = 5k$).

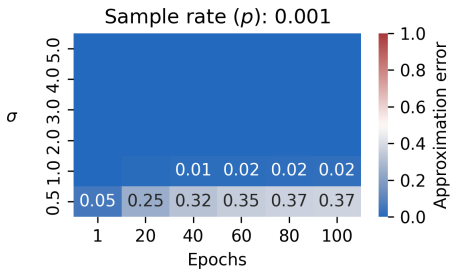

Figure 4: Approximation error between our bound (Corollary 7) and the PLD accountant, w.r.t. the noise level σ and the number of epochs (pT), with a sample rate $p = 0.001$. The error is measured as the absolute difference between the two estimates. A label for the error is shown only if it is ≥ 0.01.

Suppose that an application requires $\beta^* \geq 0.98$; assuming a uniform prior between members and non-members, this corresponds to at most a 51% attack success probability. Furthermore, suppose we wish to train for $T = 5k$ steps. We can select the noise σ and sampling rate p based on the relation:

$$p = \frac{\text{erf}^{-1}(1-\beta^*)\sqrt{2}}{\sqrt{T}}\sigma; \tag{4}$$

in this example, $p \approx 0.00035\sigma$, which guarantees the desired level of protection. Figure 3 shows this in general, for $T = 5k$.

4.1 Comparison with the PLD accountant

For the MIA threat model, there is a direct relation between Bayes security and $(0,\delta)$-DP:

Proposition 8 (Bayes security and $(0,\delta)$-LDP [6])**.** *Let $\mathcal{M} : \mathbb{S} \to \mathbb{O}$ be a randomized algorithm that is also $(0,\delta)$-LDP, and assume $\mathbb{S} = \{0,1\}$. Then:*

$$\beta^*(\mathcal{M}) = 1 - \delta.$$

Thanks to this relation, we can compare our security bounds with equivalent ones estimated via state-of-the-art (ε,δ)-DP numerical accountants. We use the PLD accountant [24, 25], which supports the substitution adjacency relationship; this matches our MIA threat model (Game 2), where the attacker has to distinguish two datasets that differ on a single record.[3]

The goal of this comparison is twofold. First, we evaluate under what parameter choices our bounds are tight (i.e., when the approximation error derived in Proposition 4 is small). Second, we compare their computational costs.

The tightness of our MIA bound. Our main result (Theorem 6) and, consequently, our bound on MIA (Corollary 7) are based on the approximation of a mixture of Gaussians to a Gaussian distribution (Proposition 4). Naturally, we do not expect this approximation to work well for all parameter choices. Based on our initial evaluation for $T = 1$ (Figure 2), we suspect it will perform better for larger values of σ.

In these experiments, we compare β^* obtained as in Corollary 7 with a $(0,\delta)$ estimate given by the PLD accountant; because PLD is tight up to discretization error, we use its estimate $\beta^* \approx 1-\delta$ as the ground truth for these experiments. Figure 4 shows the absolute error between the two estimates. We observe that our bound has a small error (≤ 0.01) for $\sigma \geq 1$ up to 50 epochs (i.e., $T = 50k$ for $p = 0.001$); as the number of epochs grows to 100, the error increases to ≈ 0.02. This shows that our bound is tight for a wide range of realistic parameters. On the other hand, we observe that for $\sigma < 1$ the error is large; intuitively this is because the approximation of a mixture with a Gaussian gets worse the smaller the variance of the Gaussian is. In practice, this means it is not advisable to use our bound for $\sigma < 1$.

Computational efficiency. We compare the costs of our bound with the PLD accountant. Figure 5 shows that our bound is orders of magnitudes faster to compute than the respective PLD estimate.[4] The time efficiency of our bound enables practitioners to select the parameters for DP-SGD interactively, in real-time, and with a high level of accuracy.

4.2 Bayes Security and (ε,δ)-DP

An important drawback of (ε,δ)-DP is that it may be harder to match to a specific threat model; in turn, this makes it difficult to select appropriate values (ε,δ). In this section, we compare Bayes security with DP in terms of their ability to capture metrics of interest for MIA. We do this analysis by relating Bayes security and (ε,δ)-DP to two quantities that are commonly used for evaluating MIA threats: the advantage,

[3] The PLD accountant also supports the add-remove adjacency relationship, which is not relevant for our threat model.

[4] Comparisons with other state-of-the-art accountants led to similar conclusions.

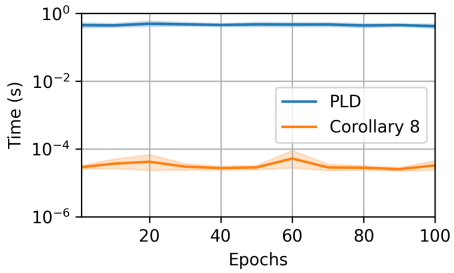

Figure 5: Computational cost of our bound (Corollary 7) and the PLD accountant, w.r.t. the number of epochs, with a sample rate $p = 0.001$. The time is measured in seconds.

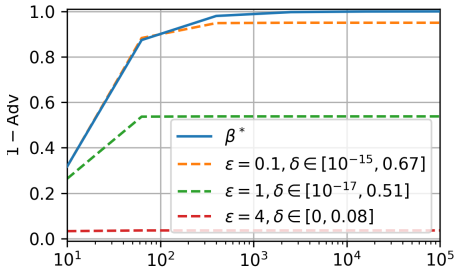

Figure 6: (ε, δ)-DP of DP-SGD. We set $N = 100k$, $L = 10$, $C = 1$, $T = 1$. Here, β^* is computed via Corollary 7.

and the true positive rate at a certain false positive rate (hereby denoted by TPR@FPR).

MIA advantage. The MIA advantage, Adv, measures how much more likely an attacker is to guess the membership of a data record having access to the trained model, compared to an attacker who only guesses based on prior knowledge. Intuitively, Adv describes the additional risk that one incurs by releasing the model, w.r.t. the MIA threat.

The equivalence between Bayes security is direct: $\beta^* = 1 - \text{Adv}$. The relation between Adv and (ε, δ)-DP was shown by Humphries et al. [20]: if a mechanism is (ε, δ)-DP, then the advantage is bounded as follows:

$$\text{Adv} \leq \frac{e^\varepsilon - 1 + 2\delta}{e^\varepsilon + 1}. \tag{5}$$

In Figure 6, we illustrate the behavior of this bound for progressively decreasing values of ε. The curves are obtained by computing (ε, δ) via the PLD accountant for DP-SGD, and then plugging them into Equation (5).

Results indicate that, by taking smaller values of ε, we get tighter bounds on the advantage. In particular, the tightest bound is achieved when $\varepsilon = 0$; this corresponds to the case when $\beta^* = 1 - \delta$. This validates the use of a notion related to $(0, \delta)$-DP: under this configuration, we can hope to achieve the tightest analysis from an advantage perspective.

TPR@FPR. Carlini et al. [5] recommended measuring the true positive rate of attacks at low false positive rates (i.e., TPR@FPR). Their reasoning is that attacks with high accuracy may be unhelpful in practice; e.g., an attack can have 99% accuracy yet not be able to identify members confidently without an unreasonable number of false positives.

To facilitate a comparison, we first prove a relation between Bayes security and TPR@FPR:

Proposition 9. *Consider a randomized mechanism $\mathcal{M} : \mathbb{S} \to \mathbb{O}$ with $\mathbb{S} = \{0, 1\}$, and let S be a random variable on $\mathbb{S}$ with $\pi = \Pr[S = 1]$. Let $s' = \text{Attacker}(\pi, \mathcal{M}(S))$ be the guess that* Attacker *makes for S given the output of the mechanism. Let the true positive rate (*TPR*) be the probability that attacker* Attacker *guesses correctly when $S = 1$, and the false positive rate (*FPR*) be the probability that they guess incorrectly when $S = 0$. If the mechanism is β^*-secure then for every attacker:*

$$\text{TPR} \leq 1 + \text{FPR} - \beta^* \quad \textit{if } \pi \leq 1/2$$
$$\text{TPR} \leq \frac{\pi}{1-\pi}(1 + \text{FPR} - \beta^*) \quad \textit{otherwise.}$$

We have equality for a uniform prior, $\pi = 1/2$.

We observe that the special case $\pi = 1/2$ was known (e.g., [34]). However, we believe the general case is novel. In practice, we expect the case $\pi \leq 1/2$ to be more relevant; for example, in MIA, the prior probability that a data record is a member is typically smaller than the alternative case.

We compare this with the bound given by f-DP, which gives the best possible bound on TPR@FPR; we remark that obtaining f-DP bounds is computationally expensive. In Figure 7, each method gives an upper bound on the TPR for a chosen FPR value. We observe that Bayes security is optimal for TPR@FPR=0.5; this matches the case when β^* is the complement of the advantage. The bound given by Bayes security becomes worse for lower levels of FPR. However, we observe a relatively small discrepancy with respect to the f-DP bound. For example Bayes security bounds TPR@0.1FPR by 0.128, while f-DP bounds it by 0.113; for a smaller FPR, Bayes security indicates TPR@0.01FPR ≤ 0.038, while the f-DP bound is 0.012.

These experiments suggest that one can use the (cheap to compute) Bayes security to obtain a good bound on TPR@FPR, and then use f-DP to tighten the bound if needed.

4.2.1 Corollary 7 as an (ε, δ)-DP Estimator

We observe that, in addition to having a direct expression for the security of DP-SGD against MIA, one could use Corollary 7 to obtain a rough estimate of (ε, δ)-DP. To this end, we can once again exploit Equation (5) by Humphries et al. [20].

By the correspondence between the advantage and Bayes security ($\text{Adv} = 1 - \beta^*$), we obtain a bound on ε as follows.

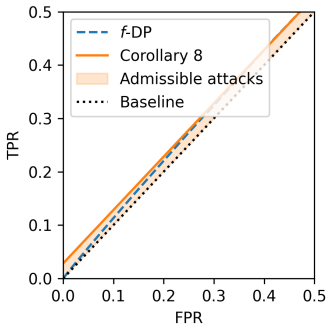

Figure 7: Bounds on TPR@FPR for DP-SGD, computed via Bayes security (Proposition 9) and f-DP. The viable region according to β^* is highlighted in orange. DP-SGD parameters: $p = 0.0001$, $N = 10k$, 50 epochs (i.e., $T = 50/p$), $\sigma = 2$.

Let β^* be the Bayes security of DP-SGD computed as per Corollary 7; then for any choice of $\delta \in [0,1)$, we get:

$$\varepsilon \geq \log - \frac{2\delta + \beta^* - 2}{\beta^*}.$$

This bound can be a cheap alternative to more computationally expensive methods (e.g. numerical accountants) for estimating ε. However, it should be remarked that this bound is loose. The inequality by Humphries et al. [20], whilst tight, applies to *any* (ε,δ)-DP algorithm: one might improve on their inequality and the above estimate for the case of DP-SGD; this is what more advanced (ε,δ)-DP estimators do.

Related work. The (ε,δ)-DP literature has explored the privacy guarantees of many basic mechanisms. In particular, Balle and Wang [2] and Sommer et al. [31] studied the privacy of a *Gaussian mechanism without subsampling*. By applying our observation that the distribution of the gradients can be approximated with a Gaussian, the special case $p = 1$ of Corollary 7 can be obtained as a consequence of their results, thanks to the relationship between Bayes security and (ε,δ)-DP. Further, the effect of subsampling in the context of DP is understood to amplify the privacy level by a factor of p [1, 28, 32]. In concurrent work, Mahloujifar et al. [27] suggested the following strategy: for a specific threat (membership inference, in their case), determine the advantage of an attacker who observes the intermediate models output by DP-SGD. Their proposal is to estimate this advantage via Monte Carlo simulations. Our analysis strategy is similar to theirs in spirit: we aim to quantify the leakage for specific threats. Differently from them, we tackle a more general case (which subsumes membership and attribute inference), and we obtain closed-form expressions for our bounds.

5 Attribute Inference

In this section, we apply Theorem 6 to measure the security of DP-SGD against AI. First, we discuss the limits of *any* security analysis: without making assumptions, one cannot improve on MIA bounds (e.g., Corollary 7). We mitigate this issue by providing data-dependent bounds for AI, and by instrumenting DP-SGD to compute them. Second, we study whether a data-dependent security analysis has any security implications. Finally, we discuss the computational time overheads of our method and ways to improve it.

5.1 Limits of *any* AI Analysis

Before stating our AI bound, it is important to understand what is achievable by a DP-SGD security analysis under this threat. As it turns out, it is impossible to obtain a non-trivial bound for AI (i.e. a bound for AI that is better than the MIA bound) without making assumptions on the *gradient function*.

This is easy to see by the following example. In AI, the attacker tries to guess the secret value S given partial information φ and the model's weights. By the arguments made in Section 3.2, we can simplify this and limit the information available to an attacker to φ and the clipped and noisy gradient: $\bar{\mathbf{g}}_t(z^*) + \text{Noise}$; further, by the arguments made in Section 3.1, we can assume there are just two secrets (i.e. $S \in \{0,1\}$). Let us now consider a contrived gradient function, which returns $\bar{\mathbf{g}}_t(\varphi \mid s_0) = -\bar{\mathbf{g}}_t(\varphi \mid s_1) = (C,0,\ldots,0)$ for every t, where C is the clipping gradient. It is easy to see that this case matches record-level MIA, and that the bound on the Bayes security against AI will be Corollary 7, which cannot be improved upon without further assumptions. Even then, it is unclear what reasonable assumptions one could make on $\bar{\mathbf{g}}_t$ without affecting the validity of a security analysis.

We address this problem by instrumenting DP-SGD to keep track of the sensitivity $\|\bar{\mathbf{g}}_t(\varphi \mid s_i) - \bar{\mathbf{g}}_t(\varphi \mid s_j)\|$, for all training points z and all possible attribute values s_i, s_j. This comes with an extra computational cost, which is although acceptable for various real-world tasks. In the next part of this section, we derive the bound on Bayes security, describe an algorithm for measuring the bound, and then describe optimization and approximation strategies.

5.2 Bayes Security of DP-SGD against AI

We can now adapt our main result (Corollary 7) to the case of AI; as before, we do this by specializing the definition of Δ_f. We write $\varphi(z^*)$ to denote the non-sensitive part of z^*.

Corollary 10. *The Bayes security of DP-SGD against AI is:*

$$\beta^*(P_{O|S}) \geq 1 - \operatorname{erf}\left(p\frac{\|R\|}{2\sqrt{2}\sigma C}\right) - O\left(\frac{\sqrt{pT}}{\sigma}\right),$$

where $R = (R_1, ..., R_T)$ *with*

$$R_t = \max_{z^* \in L_t} \max_{s_0, s_1 \in \mathcal{A}} \|\bar{\mathbf{g}}_t((\varphi(z^*), s_0)) - \bar{\mathbf{g}}_t((\varphi(z^*), s_1))\|,$$

where L_t *is the batch sampled at step* t.

Algorithm 4: AI-resilient-SGD($\{z_1, \ldots, z_N\}, \mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, z_i), \eta_t, \sigma, L, C, \mathcal{A}$)

Initialize θ_0 randomly
for $t \in [T]$ **do**
  Take a random sample L_t with sampling probability L/N
  Compute gradient
  For each $i \in L_t$, compute $\mathbf{g}_t(z_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, z_i)$
  Compute gradient bound w.r.t. attribute's value
  $R_t = \max_{z^* \in L_t} \max_{s_0, s_1 \in \mathcal{A}} \|\bar{\mathbf{g}}_t((\varphi(z^*), s_0)) - \bar{\mathbf{g}}_t((\varphi(z^*), s_1))\|$
  Clip gradient
  $\bar{\mathbf{g}}_t(z_i) \leftarrow \mathbf{g}_t(z_i) / \max\left(1, \frac{\|\mathbf{g}_t(z_i)\|_2}{C}\right)$
  Add noise
  $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(z_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
  Descent
  $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
end
Output θ_T

We describe how DP-SGD can be adapted to compute the values R_t. We observe that bounds computed in this manner are *data-dependent*: the value of R_t at step t depends on the model's parameters at that step, and on the data itself. In the next part of this section, we discuss why this has no privacy implications for the attack under consideration (AI).

Algorithm 4 modifies DP-SGD for calculating R_t. For every batch L_t and every point $z^* \in L_t$, the algorithm augments $\varphi(z^*)$ with all completions $s \in \mathcal{A}$, and determines the maximum distance between the two. The Bayes security is determined by plugging the vector $R = (R_1, ..., R_T)$ in Corollary 10.

5.3 Privacy Implications of Data-dependence

One may wonder whether computing a security metric that depends on the data may have any privacy implications; indeed, β^*, computed as per Algorithm 4, contains information about the secret. The main concern raises when revealing β^* to a malicious party: would they be able to infer any privacy information about the training set if given access to it?

We analyze this concern w.r.t. to two threat models: MIA and AI. We can describe each case similarly to Games 2-3, with the difference that the output O communicated to the attacker is the security metric β^*. Equipped with this information, the attacker's goal is to guess the secret S (i.e., *membership* of a challenge point or *attribute* value).

MIA. Revealing the Bayes security β^* of DP-SGD against AI, computed as described in Algorithm 4, *may leak* the membership of a data record. An example that supports this claim follows. Suppose the challenge points, z_0^* and z_1^*, are both such that $R_t = \max_{s_0, s_1 \in \mathcal{A}} \|\bar{\mathbf{g}}_t((\varphi(z_b^*), s_0)) - \bar{\mathbf{g}}_t((\varphi(z_b^*), s_1))\|$; the parameter R_t^b, computed for either challenge point z_b^*, is maximized by that challenge point. Further, suppose $R_t^0 \neq R_t^1$. In this special case, the attacker can infer the challenge point from β^*. Based on this observation, we recommend that whenever both MIA and AI are a concern, the security parameter estimated for the AI analysis is not revealed to the public.

AI. If the main concern is AI, no information is revealed by β^* itself. The reason for this is that R_t is computed based on all possible values $\mathcal{A}$ for the attribute. Therefore, R_t (hence, β^*) is the same regardless of the value of the sensitive attribute s. Therefore, if AI is the only threat of concern for a deployment, it is safe to reveal the estimated security metric to the public.

5.4 Computational Costs and Optimization

The cost of Algorithm 4 grows quadratically in the number of attributes $|\mathcal{A}|$. In our experiments (Section 6), we observe that the time overhead is acceptable for small attribute spaces. Nevertheless, as $\mathcal{A}$ grows, this cost becomes too high. We explore two strategies for reducing this cost.

Domain knowledge. As a first strategy, we can use the fact that Bayes security is maximized over two secret values only (Theorem 1); in particular, these should be the two values $s_0, s_1 \in \mathcal{A}$ that maximize the attacker's advantage. In many practical applications, we can exploit domain knowledge to decide in advance what values will likely give the attacker the best advantage. For example, consider the MNIST dataset, where each data record is a pixel matrix, and where each pixel is represented by a value in $[0, 1]$. Suppose the sensitive attribute is one of such pixels.[5] In this case, we can make the assumption that the two values maximizing the risk for the attribute will be the two extremes, $\{0, 1\}$. This observation can substantially reduce the computation cost of Algorithm 4.

The point set diameter problem. A second strategy is to approximate the value R_t. To this end, we observe that finding the distance between the two maximally distant gradients is an instance of the well-known point set diameter problem, which is defined as follows. Let (M, d) be a metric space on a finite set M for some metric d. A solution to the point set diameter problem is an algorithm that returns $\text{diam}_M = \max_{x,y \in M} d(x, y)$. Various exact and approximate solutions exist for this problem [6, 21, 33]. In this paper, we consider a simple $O(N)$ solution, where $N = |M|$, which gives a lower

[5] We could equivalently define the risk for a set of pixels at once. A similar argument would apply.

bound based on the triangle inequality: for any choice $x \in M$, we have $\text{diam}_M \leq 2 \max_{y \in M} d(x,y)$.

Let v be the mean vector of the gradients $\{\bar{\mathbf{g}}_t((\varphi(z^*),a))\}_{a \in \mathcal{A}}$; we estimate R_t as:

$$R_t \leq 2 \max_{a \in \mathcal{A}} \|\bar{\mathbf{g}}_t((\varphi(z^*),a)) - v\| .$$

Naturally, the choice of a *lower* bound here is security-motivated: it measures the worst-case for the victim.

Note that this estimate can be improved either by picking more carefully the point v, or by running this algorithm for various choices of v and then choosing the one giving the tightest bound. Despite the approximation given by the triangle inequality, in our experiments we observed this approximation to be good enough. Nevertheless, practical applications may consider solutions that give tighter bounds (e.g., [21]).

Related and Future work. With an appropriate choice of adjacency relationship, one can capture the AI threat in DP. One may wonder whether this observation enables adapting accountant-based analyses of DP-SGD to this threat. We observe that DP-SGD uses gradient-clipping to bound sensitivity. Since gradients are computed per-example, there is no stronger data-independent sensitivity bound for AI than MIA when adapting neighboring datasets as differing in one attribute in one record [22]. Unfortunately, attribute-DP mechanisms [37] are impractical for ML.

Opportunities for future work include further improvements to the computational efficiency of our AI bounds analysis. When using the approximate algorithm, the bottleneck of the analysis becomes computing the gradients of data records obtained by replacing their sensitive attribute. A promising strategy is to use influence functions (IF) [10, 23] to approximate this more efficiently. The main idea behind IF is to *efficiently* approximate the addition and removal of a training point to a trained model via a Taylor approximation of a Newton step. We observe that future work may explore further strategies. In addition to using alternative solutions to the point set diameter problem, one could use optimization algorithms such as gradient descent to obtain the value of R_t more quickly. Future work may also explore approximations of the R_t expression, e.g. by using a Newton approximation by taking inspiration from the influence functions literature.

6 Empirical evaluation

We evaluate our security analyses on models trained via DP-SGD on two datasets. First, we study the computational costs of the AI analysis, and the effectiveness of its approximation (Section 5). Second, we compare the privacy-utility trade-offs that our MIA and AI analyses can offer.

Datasets. We use two tabular datasets; this makes it meaningful to conduct an attribute inference analysis. They are the Adult Census Income dataset (`Adult`) and the Purchase dataset (`Purchase`). The `Adult` dataset has 32,561 records with 108 attributes each (after one-hot-encoding). It contains data from the 1994 US Income Census, and the learning task is to predict the income of a person (precisely, whether it is above 50K/year or not), given attributes such as age and education. Importantly, it has attributes taking more than 2 possible values; this facilitates a comparison between the "full" and "approximate" AI analyses. We select *age* to be the sensitive attribute for the AI analysis: this attribute has 73 unique values, ranging between 17 and 90. For the purpose of the AI analysis, we consider the entire range $\{17, 18, ..., 90\}$.

The `Purchase` dataset has 197,324 records and 600 attributes. Each record correspond to one customer, and each (binary) attribute indicates whether the customer bought a particular item. This dataset enables evaluating how well our analyses scale to larger datasets. For the AI analysis, we select the first attribute (purchase) to be the sensitive one.

Models and setup. We train two fully connected neural networks as described by Bao et al. [3], implemented via `pytorch`. We instrumented `Opacus` [35] to support our AI analysis, as a callback function that is run at every step.

Privacy parameters. We use Equation (4) for selecting the privacy parameters p and σ. For illustration purposes, we aim at a MIA Bayes security $\beta^* = 0.9$ after 20 epochs; this sub-optimal security against MIA enables observing the benefits of the AI analysis. We run DP-SGD for 30 epochs; this enables observing the behavior of β^* after the predicted number of epochs. We let $L = 256$ for the `Adult` dataset, and 512 for the `Purchase` dataset. This, paired with the training set size, enables determining the following noise parameters (Equation (4)): $\sigma = 3.51$ (`Adult`) and $\sigma = 1.8$ (`Purchase`).

6.1 AI analysis

Running Time. First, we evaluate the computational costs of our analysis. We present the measurements for the `Adult` dataset, which is harder to tackle from an AI analysis perspective; indeed, for each step we need to compute $N \times |\mathcal{A}|$ gradients, where N is the size of the training set, and $\mathcal{A}$ are the possible values for the sensitive attribute; further, for every step we need to solve the point set diameter problem for the set of generated gradients, which is expensive (Section 5).

We train the model enabling one of the following analyses: DP accountant, MIA, AI (approximate), AI (full). As a baseline, we include the training time for the same model without DP. We run this for 10 epochs (26 steps per epoch). Figure 8 shows the average time taken to train for one epoch.

The cost of training with DP is roughly twice the cost of training without. We can see that running our MIA analysis gives a rather marginal improvement over a DP accountant; indeed, we expect that the computational advantages of our

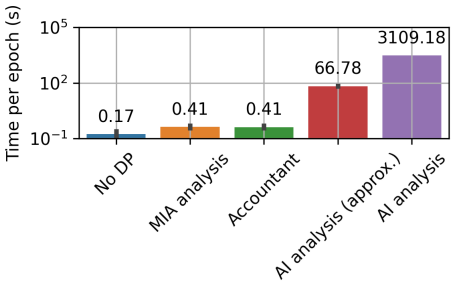

Figure 8: Average running time per epoch, across 10 epochs. `Adult` dataset (sensitive attribute has 73 possible values).

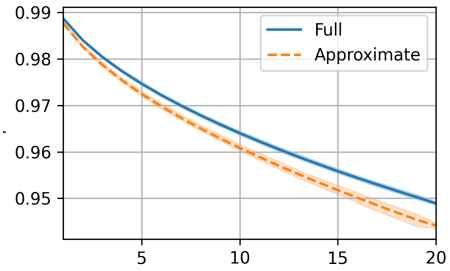

Figure 9: Bayes security on the `Adult` dataset, estimated via the full and approximate AI analysis.

analysis (Section 4) would prove more useful during parameter search than during training. Unsurprisingly, we can see that both AI analyses have large costs relatively to the others. In Section 6.2, we observe that this may be a fair price to pay given their advantages from a privacy-utility perspective.

Finally, the approximate analysis reduces the costs by one order of magnitude. In the next part, we measure how much this approximation affects the quality of the analysis.

Full vs Approximate AI Analysis. In the previous part, we observed that the approximate AI analysis heavily reduces the computational costs. Figure 9 compares the Bayes security estimates of the two analyses. These experiments are run on the `Adult` dataset, whose sensitive attribute has more than 2 values; this enables appreciating the difference between the approximate and full version. We remark that the approximate analysis can never indicate more security than the true one: it is a lower bound of the full AI analysis by construction. The results suggest that the price one has to pay when running the approximate AI analysis as opposed to the full one is fairly small. The estimated bounds on Bayes security after 20 epochs are $\beta^* \approx 0.945$ for the full analysis and $\beta^* \approx 0.950$ for the approximate analysis.

Overall, while the costs of the approximate AI analysis are much larger than standard DP-SGD training, it is still a relatively scalable way of training, and it has advantages in the privacy-utility tradeoff, as illustrated in the next section.

6.2 Bayes Security of DP-SGD

Figure 1 relates the Bayes security (MIA and AI) with the accuracy of the trained model; to avoid jitters, we round the accuracy to the closest multiple of 10%, and show the corresponding confidence region. For reference, we include the (ε, δ)-DP estimate provided by a numerical accountant.

Comparison with (ε, δ)-DP. We observe that, for the chosen parameters, the risk of MIA is relatively high even for low values of ε: for $\varepsilon \approx 0.5$, the risk of MIA is $\beta^* \approx 0.9$ (`Adult`). By the relation between Bayes security and the attacker accuracy, and assuming a uniform prior on the membership, this value of β^* implies that the attacker can guess the membership correctly with 55% probability. Similarly, for an attacker aiming for a maximum FPR or 10%, their TPR is at most 20% (`Adult`). This is a relatively high risk, considering that it matches a value of ε that would generally be considered to be extremely safe. Of course, this analysis depends on various assumptions: i) that the attacker knows the entire training set minus the challenge point, ii) that they can access all the gradient updates, and iii) that the MIA threat model is a concern for the particular deployment. We now show empirically how relaxing the last assumption offers important privacy-utility benefits; we discuss the other two assumptions in Section 7.

Empirical comparison between threat models. A principled way of relaxing a security analysis is to adopt a weaker threat model; e.g., there are practical applications where MIA may not be a concern, but AI is.

Consider the best models trained on `Adult` (83% accuracy) and `Purchase` (73% accuracy). We compare their resilience against MIA and AI by mapping their Bayes security to i) the attacker's success rate, and ii) TPR at FPR$\leq$10%. To compute the former, we need to assume a prior on the membership (resp., values of the sensitive attribute); for simplicity, we assume a uniform prior. As for TPR@FPR, we notice it is not well-defined in general for AI. We define it as follows: assume that there is a particularly damaging value of the sensitive attribute, which the attacker aims at predicting precisely; for example, in the `Purchase` dataset, the attacker may be interested in the value '1', corresponding to a product purchase. We define TPR@FPR by considering this value to be the "positives" class. The table below summarizes the results; we denote the attacker's success rate by V.

Task	β^* AI (MIA)	**V** AI (MIA)	**TPR** AI (MIA)
`Adult`	0.93 (0.88)	53% (56%)	$\leq$ 16% (20%)
`Purchase`	0.99 (0.87)	51% (57%)	$\leq$ 11% (23%)

We observe that, for both tasks, the AI analysis gives

a much tighter bound on the attacker's probability of success, and on the TPR@FPR, than the MIA analysis. In particular, we observe that best performing model on the `Purchase` dataset, while relatively unsecure against MIA (e.g., TPR$\leq$23%), is almost perfectly secure against AI (TPR$\leq$11%). This emphasizes the benefits of a more nuanced analysis of the privacy risks of DP-SGD, which takes into account the attacker's knowledge and goals.

7 Towards Inference-time Attackers

We apply our formalization to capture an *inference-time* attacker, and discuss obstacles and future directions.

Training-time attacker. Both DP-guided analyses and our techniques have an important underlying assumption: that *the attacker is able to inspect (and, possibly, modify) the intermediate gradients produced during training*. This *training-time* attacker is widely used throughout the literature, and has led to state-of-the-art results, such as the tight bounds for MIA obtained using the PLD accountant or the approach by Humphries et al. [20]. However, this assumption is quite strong in practice. Whilst it might hold in a federated learning setting, where the attacker can inspect (and possibly modify) the gradients during training, it it not representative of settings in which the adversary cannot observe the training process, e.g., a model trained privately in a secure environment.

Inference-time attacker. In contrast to the training-time attacker, the *inference-time* attacker cannot observe the training process, and only has access to the final model. Note that we can still consider the standard white-box vs. black-box duality for the inference-time attacker, which refers to whether the attacker has either full access to the model's weights or only the ability to query the model and receive responses. Intuitively, the inference-time attacker has less information about the model than the training-time attacker, so it is reasonable to assume that the former should be weaker than the latter. Evaluating security against an inference-time attacker may therefore yield significantly better bounds.

Modelling inference-time attackers with our framework. We expect our analysis techniques can be used for studying this weaker (albeit more realistic) threat scenario. Formally, this is modelled by an attacker who tries to guess secret S given only the final model weights O_T; note that the secret S would be left unchanged. As for deriving the bounds, one may be tempted to use a similar strategy to the one we used in Section 3. Unfortunately, we found this to be non-trivial.

The main difficulty in applying our results to this problem comes from the effect that the challenge point z^*, (possibly) sampled at step t, has on the subsequent gradient functions $\bar{\mathbf{g}}_{t+1}(\cdot), \bar{\mathbf{g}}_{t+2}(\cdot), \ldots$. In our analysis (Section 3.2), we could disregard the effect of $\bar{\mathbf{g}}_t(z)$ for points $z \neq z^*$ for $t > 1$. The reason is that $\bar{\mathbf{g}}_t(z)$ did not bear *additional* information about the challenge point to the adversary (who has access to $\bar{\mathbf{g}}_{t-1}(z)$ as well). This reasoning does not apply to an inference-time attacker: since $\bar{\mathbf{g}}_t(\cdot)$ might have seen the effect of z^* in a previous step, and the attacker does not know $\bar{\mathbf{g}}_{t-1}(\cdot)$, we cannot disregard it as a potential source of leakage.

Explored directions. One way to approach this problem is via a taint analysis. Let p be the probability that the challenge point z^* is sampled at step t. Then we can write a combinatorial expression that factors the likelihood that gradients (at points $z \neq z^*$) were tainted by the presence of z^*. Unfortunately, this approach does not improve substantially on the analysis that we discussed in this paper: since DP-SGD is usually run for a number of steps that is proportional to the number of batches, the probability "z^* was sampled before step t" grows exponentially with t; leading to trivial bounds.

We suspect that to obtain a tighter analysis of the security of DP-SGD against inference-time attackers, one may need to quantify the *influence* of the challenge point on the gradient function. Influence functions [18, 23] may be a good tool to approximate this; however, it should be noted, this would require making assumptions about the gradient function. An alternative is to use black-box leakage estimation methods (e.g., [9]), which however require an infinite amount of data to provably convergence [8, Theorem 2.7].

8 Conclusion

The privacy of DP-SGD has historically been measured via DP-guided moment accountants. This practice comes with various issues: i) computational complexity, ii) accountants are hard to implement correctly [15, 17, 24], iii) and DP is traditionally applied in a threat-agnostic manner.

Our proposal gives closed-form bounds for the privacy of DP-SGD, which are both easy to implement and orders of magnitude faster to compute than state-of-the-art DP estimators. Additionally, our bounds are *threat-specific*. This has two main benefits: i) they are arguably more interpretable, as they capture the risk for each threat individually; and ii) in circumstances where a weaker threat model (e.g., AI) is acceptable, one can achieve a much better utility at the same privacy level (Figure 1). Finally, our bound on the resilience against the AI threat model is *data-dependent*; this enables further pushing the utility depending on the inherent leakage of the data itself. Given the simplicity of modelling training algorithms (and respective threats) as information theoretic channels, we expect our analysis strategy can be used to derive bounds for other threats for existing training algorithms, or new ones designed with this strategy in mind.

References

[1] Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *23rd ACM SIGSAC Conference on Computer and Communications Security, CCS 2016*, pages 308–318. ACM, 2016. doi: 10.1145/2976749.2978318.

[2] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pages 394–403. PMLR, 2018.

[3] Wenxuan Bao, Luke A Bauer, and Vincent Bindschaedler. On the importance of architecture and feature selection in differentially private machine learning. *arXiv preprint arXiv:2205.06720*, 2022.

[4] SS Barsov and Vladimir V Ul'yanov. Estimates of the proximity of Gaussian measures. *Sov. Math., Dokl*, 34: 462–466, 1987.

[5] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[6] K. Chatzikokolakis, G. Cherubin, C. Palamidessi, and C. Troncoso. Bayes security: A not so average metric. In *2023 2023 IEEE 36th Computer Security Foundations Symposium (CSF) (CSF)*, pages 159–177. IEEE Computer Society, jul 2023. doi: 10.1109/CSF57540.2023.00011.

[7] Giovanni Cherubin. Bayes, not naïve: Security bounds on website fingerprinting defenses. *Proceedings on Privacy Enhancing Technologies*, 2017.

[8] Giovanni Cherubin. *Black-box security: measuring black-box information leakage via machine learning*. PhD thesis, Royal Holloway, University of London, 2019.

[9] Giovanni Cherubin, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. F-BLEAU: fast black-box leakage estimation. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 835–852. IEEE, 2019.

[10] R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4): 495–508, 1980.

[11] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[12] Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The total variation distance between high-dimensional Gaussians with the same mean. *arXiv preprint arXiv:1810.08693 [math.ST]*, 2018. doi: 10.48550/ARXIV.1810.08693.

[13] Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy, 2019. URL https://arxiv.org/abs/1905.02383.

[14] Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions, 2022. URL https://arxiv.org/abs/2207.04380.

[15] Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions. *arXiv preprint arXiv:2207.04380*, 2022.

[16] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1322–1333. ACM, 2015. doi: 10.1145/2810103.2813677.

[17] Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642, 2021.

[18] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

[19] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE, 2007.

[20] Thomas Humphries, Simon Oya, Lindsey Tulloch, Matthew Rafuse, Ian Goldberg, Urs Hengartner, and Florian Kerschbaum. Investigating membership inference attacks under data dependencies. *CoRR*, abs/2010.12112v3, 2021.

[21] Mahdi Imanparast, Seyed Naser Hashemi, and Ali Mohades. Efficient approximation algorithms for point-set diameter in higher dimensions. *Journal of Algorithms and Computation*, 51(2):47–61, 2019.

[22] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204, 2011.

[23] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[24] Antti Koskela and Antti Honkela. Computing differential privacy guarantees for heterogeneous compositions using fft, 2021. URL https://arxiv.org/abs/2102.12412.

[25] Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using fft. 2020. doi: 10.48550/ARXIV.2006.07134. URL https://arxiv.org/abs/2006.07134.

[26] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. Membership privacy: A unifying framework for privacy definitions. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS*, page 889–900. ACM, 2013. doi: 10.1145/2508859.2516686.

[27] Saeed Mahloujifar, Alexandre Sablayrolles, Graham Cormode, and Somesh Jha. Optimal membership inference bounds for adaptive composition of sampled gaussian mechanisms. *arXiv preprint arXiv:2204.06106*, 2022.

[28] Ilya Mironov, Kunal Talwar, and Li Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[29] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, S&P*, pages 3–18. IEEE, 2017. doi: 10.1109/SP.2017.41.

[30] Geoffrey Smith. On the foundations of quantitative information flow. In *International Conference on Foundations of Software Science and Computational Structures*, pages 288–302. Springer, 2009.

[31] David Sommer, Sebastian Meiser, and Esfandiar Mohammadi. Privacy loss classes: The central limit theorem in differential privacy. *Cryptology ePrint Archive*, 2018.

[32] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

[33] Andrew Chi-Chih Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.

[34] Samuel Yeom, Irene Giacomelli, Alan Menaged, Matt Fredrikson, and Somesh Jha. Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning. *Journal of Computer Security*, 28(1):35–70, 2020. doi: 10.3233/JCS-191362.

[35] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

[36] Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Ahmed Salem, Victor Rühle, Andrew Paverd, Mohammad Naseri, and Boris Köpf. Bayesian estimation of differential privacy. *arXiv preprint arXiv:2206.05199*, 2022.

[37] Wanrong Zhang, Olga Ohrimenko, and Rachel Cummings. Attribute privacy: Framework and mechanisms. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 757–766, 2022.

A Proofs

Proposition 4. *Let $f_{\mathcal{M}}$ be a Gaussian mixture defined as follows. For a mean vector $\mu = (\mu_1, ..., \mu_T)$ and covariance matrix $\sigma^2 C^2 \mathbf{I}_T$, and $C = \max_{j=1}^{T} \mu_j$, let $f_{\mathcal{M}}(x) = \sum_{b\in\{0,1\}^T} \pi_b f_{\mathcal{N}(\mu b, \sigma^2 C^2)}(x)$. The i-th component takes values from $f_{\mathcal{N}(\mu_i, \sigma^2 C^2)}$ with probability $p \in [0,1]$, or from $f_{\mathcal{N}(0, \sigma^2 C^2)}$ otherwise. Here, $\pi_b = p^{|b|}(1-p)^{T-|b|}$. The error committed in approximating $f_{\mathcal{M}}$ with $f_{\mathcal{N}(p\mu, \sigma^2 C^2)}$ is:*

$$\mathrm{tv}(f_{\mathcal{M}}, f_{\mathcal{N}(p\mu, \sigma^2 C^2)}) = O\left(\frac{\sqrt{pT}}{\sigma}\right)$$

Proof. First, observe that the total variation distance is related to the KL divergence D_{KL} as follows: $\mathrm{tv}(p_S, q_S) \leq \sqrt{\frac{1}{2} D_{KL}(p_S, q_S)}$.

We use the following bound on the KL divergence between two Gaussian mixtures (see Cover [11] and Eq. 13 in [19]):

$$D_{KL}(f_{\mathcal{M}}, f_{\mathcal{N}(p\mu, \sigma^2 C^2)}) \leq \sum_{b\in\{0,1\}^T} \pi_b D_{KL}(f_{\mathcal{N}(\mu b, \sigma^2 C^2)}, f_{\mathcal{N}(p\mu, \sigma^2 C^2)}),$$

where the KL divergence between two d-variate Gaussians, respectively centered in μ_0 and μ_1, is:

$$D_{KL}(f_{\mathcal{N}(\mu_0, \sigma_0^2)}, f_{\mathcal{N}(\mu_1, \sigma_1^2)}) = \frac{1}{2}\big((\mu_0 - \mu_1)^\intercal \Sigma_1^{-1}(\mu_0 - \mu_1) + \mathrm{tr}(\Sigma_1^{-1}\Sigma_0) - \ln\frac{|\Sigma_0|}{|\Sigma_1|} - T\big);$$

Observe that, for $\Sigma_0 = \Sigma_1 = \sigma^2 \mathbf{I}_T$, we have $\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) - \ln \frac{|\Sigma_0|}{|\Sigma_1|} - T = 0$. From the above, we obtain:

$$\begin{aligned}
D_{KL}(f_{\mathcal{M}}, f_{\mathcal{N}(p\mu,\sigma^2C^2)}) &\leq \sum_{b\in\{0,1\}^T} \frac{\pi_b}{2\sigma^2C^2}\left(\sum_{i=1}^{T}\mu_i^2(b_i-p)^2\right) \\
&\leq \sum_{b\in\{0,1\}^T} \frac{\pi_b}{2\sigma^2C^2}\left(\sum_{i=1}^{T}\max_{j=1}^{T}\mu_j^2(b_i-p)^2\right) \\
&= \sum_{b\in\{0,1\}^T} \frac{\pi_b}{2\sigma^2}\left(\sum_{i=1}^{T} b_i^2 + p^2T - 2p\sum_{i=1}^{T} b_i\right) \\
&= \sum_{b\in\{0,1\}^T} \frac{\pi_b}{2\sigma^2}\left(p^2T + (1-2p)\sum_{i=1}^{T} b_i\right) \\
&= \frac{1}{2\sigma^2}\left(p^2T\sum_{b\in\{0,1\}^T}\pi_b + (1-2p)\sum_{b\in\{0,1\}^T}\pi_b|b|\right) \\
&= \frac{1}{2\sigma^2}\left(pT - p^2T\right)
\end{aligned}$$

We used the fact that $b_i^2 = b_i$. □

For a fixed T, the goodness of this approximation depends on the choices of σ and of the sampling rate p. In our main result, T is the number of DP-SGD steps. The result matches expectations: if one wants to run DP-SGD for longer and retain strong security, one either needs to increase the noise multiplier or reduce the sampling rate.

In Figure 2, we compare the approximation error between the two distributions for a varying ratio between noise and sampling rate parameters. We observe that Proposition 4 holds when the ratio is small. In particular, for realistic regimes with $p/\sigma < 10^{-3}$, we observe a negligible approximation error ($< 10^{-4}$). In Section 4, we observe that these values for the parameters are not only practical; they are recommended to achieve stronger levels of security against MIA threats.

Theorem 6. *Assume that f is a bijection, and let Δ_f be defined as in Equation (3). The Bayes security of DP-SGD with respect to the record-level property inference threat described in Section 3 is:*

$$\beta^*(P_{O|S}) \geq 1 - \mathrm{erf}\left(p\frac{\Delta_f}{2\sqrt{2}\sigma C}\right) - O\left(\frac{\sqrt{pT}}{\sigma}\right).$$

Proof. By Corollary 2, the relation between Bayes security and the total variation Proposition 3:

$$\begin{aligned}
\beta^*(P_{O|S}) &\geq \beta^*(P_{G|S}) \\
&= 1 - \max_{s_0,s_1\in\mathrm{codom}(f)} \mathrm{tv}(P_{G|S=s_0}, P_{G|S=s_1})
\end{aligned}$$

We now determine the total variation term. Observe the following basic consequence of the triangle inequality. Let ν_a and ξ_a be two distributions parameterized by $a \in \{0,1\}$. Then:

$$\mathrm{tv}(\nu_0,\nu_1) \leq \mathrm{tv}(\xi_0,\xi_1) + \mathrm{tv}(\nu_0,\xi_0) + \mathrm{tv}(\nu_1,\xi_1).$$

We use this to replace the Gaussians mixture $P_{G|S}$ with a Gaussian, replacing the two pairwise distances $\mathrm{tv}(\nu_0,\xi_0), \mathrm{tv}(\nu_1,\xi_1)$ with an error term as defined in Proposition 4.

For any two $s_0, s_1 \in \mathrm{codom}(f)$, we have:

$$\begin{aligned}
&\mathrm{tv}(P_{G|S=s_0}, P_{G|S=s_1}) \\
&= \mathrm{tv}\Big(\sum_{b\in\{0,1\}^T} c_b P_{G|B=b,S=s_0}, \sum_{b\in\{0,1\}^T} c_b P_{G|B=b,S=s_1}\Big) \\
&\leq \mathrm{tv}(\mathcal{N}(p\bar{\mathbf{g}}(f^{-1}(s_0)),\sigma^2C^2), \mathcal{N}(p\bar{\mathbf{g}}(f^{-1}(s_1)),\sigma^2C^2)) \\
&\quad + O\left(\frac{\sqrt{pT}}{\sigma}\right) \\
&= \mathrm{erf}\left(p\frac{\|\bar{\mathbf{g}}(f^{-1}(s_0)) - \bar{\mathbf{g}}(f^{-1}(s_1))\|_F}{2\sqrt{2}\sigma C}\right) + O\left(\frac{\sqrt{pT}}{\sigma}\right)
\end{aligned}$$

In the first step, we used the above consequence of the triangle inequality. In the second step, we used the fact that $P_{G|B=b,S}$ is a Gaussian distribution and applied Corollary 5. □

Proposition 9. *Consider a randomized mechanism $\mathcal{M} : \mathbb{S} \to \mathbb{O}$ with $\mathbb{S} = \{0,1\}$, and let S be a random variable on $\mathbb{S}$ with $\pi = \Pr[S=1]$. Let $s' = \mathrm{Attacker}(\pi, \mathcal{M}(S))$ be the guess that* Attacker *makes for S given the output of the mechanism. Let the true positive rate (*TPR*) be the probability that attacker* Attacker *guesses correctly when $S = 1$, and the false positive rate (*FPR*) be the probability that they guess incorrectly when $S = 0$. If the mechanism is β^*-secure then for every attacker:*

$$\begin{aligned}
&\mathrm{TPR} \leq 1 + \mathrm{FPR} - \beta^* \quad \textit{if } \pi \leq 1/2 \\
&\mathrm{TPR} \leq \frac{\pi}{1-\pi}(1 + \mathrm{FPR} - \beta^*) \quad \textit{otherwise.}
\end{aligned}$$

We have equality for a uniform prior, $\pi = 1/2$.

Proof. Chatzikokolakis et al. [6] show that Bayes security is the minimum of the ratio between the probability that the attacker guesses *incorrectly* having access to the mechanism, $Pr[\mathrm{Attacker}(\pi, \mathcal{M}(S)) \neq S]$, and the probability that the attacker guesses incorrectly without access to the mechanism, $Pr[\mathrm{Attacker}(\pi) \neq S]$; that is:

$$\beta^* \leq \frac{Pr[\mathrm{Attacker}(\pi, \mathcal{M}(S)) \neq S]}{Pr[\mathrm{Attacker}(\pi) \neq S]} \quad \forall \pi \in (0,1).$$

We rewrite the above in terms of TPR and FPR. Say the mechanism is run k times, each time for a secret S sampled according to the prior distribution π. Let FP and TP be the

count of false positives and false negatives across these k trials. Then:

$$Pr[\text{Attacker}(\pi, \mathcal{M}(S)) \neq S] = \frac{\text{FP}+\text{FN}}{k}$$
$$Pr[\text{Attacker}(\pi) \neq S] = \frac{\min(\text{P},\text{N})}{k}$$

By combining the above, we have:

$$\begin{aligned}\beta^* &\leq \frac{Pr[\text{Attacker}(\pi, \mathcal{M}(S)) \neq S]}{Pr[\text{Attacker}(\pi) \neq S]} \\ &= \frac{k}{\min(\text{P},\text{N})}\left(\frac{\text{FP}+\text{FN}}{k}\right) \\ &= \frac{\text{P}}{\min(\text{P},\text{N})}(\text{FPR}+(1-\text{TPR}))\end{aligned}$$

where $\text{P} = \pi k$ and $\text{N} = (1-\pi)k$ are the number of positive and negative samples, respectively. The proof is concluded by considering separately the cases $\text{P} \leq \text{N}$ and $\text{P} > \text{N}$. □

We now discuss the proofs for Corollary 7 and Corollary 10.

Corollary 7. *The Bayes security of DP-SGD against record-level MIA (Game 2) is:*

$$\beta^* \geq 1 - \text{erf}\left(p\frac{\sqrt{T}}{\sqrt{2}\sigma}\right) - O\left(\frac{\sqrt{pT}}{\sigma}\right).$$

Proof. Observe that in the MIA threat model (Game 2) $f(z_s^*) = s$ is a bijection. Then:

$$\begin{aligned}\Delta_f &= \max_{z_0^*, z_1^* \in D} \|\bar{\mathbf{g}}(z_0^*) - \bar{\mathbf{g}}(z_1^*)\|_F \\ &= \max_{z_0^*, z_1^* \in D} \sqrt{\sum_{t=1}^{T} \|\bar{\mathbf{g}}_t(z_0^*) - \bar{\mathbf{g}}_t(z_1^*)\|^2} \\ &\leq 2C\sqrt{T}.\end{aligned}$$

Applying Theorem 6 concludes the proof. □

Corollary 10. *The Bayes security of DP-SGD against AI is:*

$$\beta^*(P_{O|S}) \geq 1 - \text{erf}\left(p\frac{\|R\|}{2\sqrt{2}\sigma C}\right) - O\left(\frac{\sqrt{pT}}{\sigma}\right),$$

where $R = (R_1, ..., R_T)$ *with*

$$R_t = \max_{z^* \in L_t} \max_{s_0, s_1 \in \mathcal{A}} \|\bar{\mathbf{g}}_t((\varphi(z^*), s_0)) - \bar{\mathbf{g}}_t((\varphi(z^*), s_1))\|,$$

where L_t *is the batch sampled at step* t.

Proof. Observe that f is a bijection: as per Game 3, for every z^*, there is exactly one value $s \in \mathcal{A}$ such that $f(z^*) = s$. We bound Δ_f as defined in Equation (3):

$$\begin{aligned}\Delta_f &\leq \max_{s_0, s_1 \in \mathcal{A}, z^* \in D} \|\bar{\mathbf{g}}(\varphi(z^*) \mid s_0) - \bar{\mathbf{g}}(\varphi(z^*) \mid s_1)\|_F \\ &= \max_{s_0, s_1 \in \mathcal{A}, z^* \in D} \sqrt{\sum_{t=1}^{T} \|\bar{\mathbf{g}}_t(\varphi(z^*) \mid s_0) - \bar{\mathbf{g}}_t(\varphi(z^*) \mid s_1)\|^2} \\ &\leq \sum_{t=1}^{T} \max_{s_0, s_1 \in \mathcal{A}, z^* \in D} \|\bar{\mathbf{g}}_t(\varphi(z^*) \mid s_0) - \bar{\mathbf{g}}_t(\varphi(z^*) \mid s_1)\|\end{aligned}$$

We then apply Theorem 6. Note that in this corollary's statement, we range $z^* \in L_t$, where L_t is the batch sampled at time t. This is allowed by observing that, if z^* is not included in the batch at step t, it cannot influence the model weights (and gradients) at that step. □

B Tightness of MIA bound (Corollary 7)

We report results for further sample rates in Figure 10. Results for $p = 0.001$ (Figure 4) are reported again, for comparison. We observe that our observations hold for these parameters: the approximation error is small when $\sigma \geq 1$; in general, a larger sampling rate p may further increase this error, but not significatively.

We also note that the PLD accountant failed to provide a bound for $p = 0.0001$ and $\sigma = 0.5$. Further, we observe that for the same sample rate, the approximation increases for a larger σ. We attribute this to numerical errors in the PLD accountant.

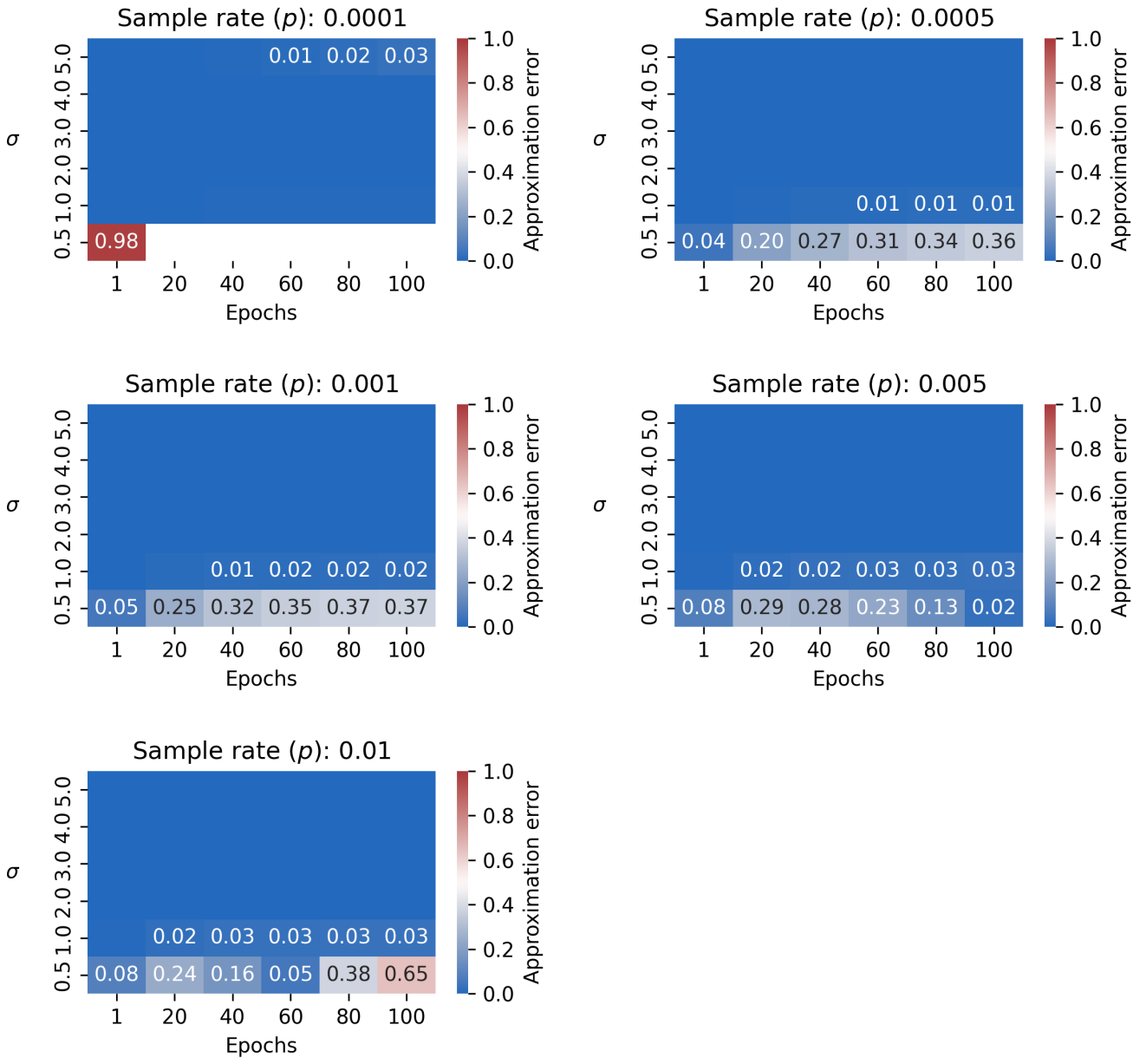

Figure 10: Approximation error of the PLD accountant for different sample rates p and noise multipliers σ.