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

Giovanni Cherubin
MSRC, Azure Research

Boris Köpf
Azure Research

Andrew Paverd
MSRC, Azure Research

Shruti Tople
Azure Research

Lukas Wutschitz
M365 Research

Santiago Zanella-Béguelin
Azure Research

Abstract

Machine learning models trained with differentially-private (DP) algorithms such as DP-SGD enjoy resilience against a broad class of privacy attacks. While one can derive bounds for some attacks solely from an (ε, δ)-DP guarantee, meaningful bounds require a small enough privacy budget (i.e. injecting a large amount of noise), which translates into 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 resorting to the threat-agnostic notion of DP. Our method focuses on the popular algorithm DP-SGD, and derives simple closed-form bounds against these privacy threats. Our proof technique is based on modelling DP-SGD as an information theoretic channel, whose inputs are the secret information an attacker is trying to guess (e.g., membership of a data record), and whose outputs are the intermediate model weights produced by the algorithm. We obtain bounds for membership inference that match state-of-the-art techniques, albeit being orders of magnitude faster to compute. Additionally, we present novel data-dependent bounds against attribute inference that allow gaining further utility from the model without an impact on privacy. Our results provide a more direct, interpretable, and practical way to evaluate the privacy of trained models against specific privacy threats, without sacrificing accuracy or performance.

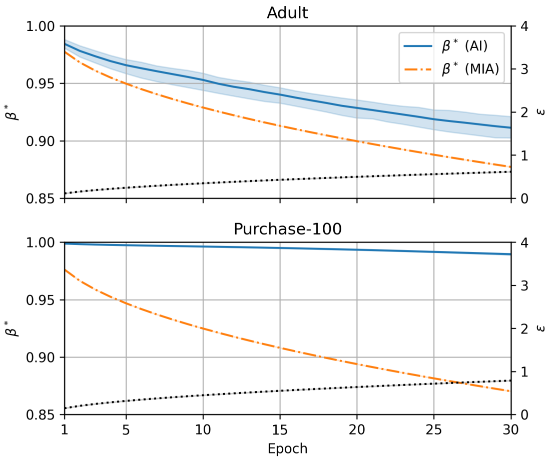


Figure 1: Bayes security of DP-SGD against MIA and AI on the `Adult` (top) and `Purchase100` (bottom) datasets. For reference, we report the (ε, δ)-DP guarantee of a numerical accountant (dashed black line), for $\delta = 3.8 \times 10^{-6}$ (`Adult`) and $\delta = 4 \times 10^{-7}$. A model is more secure (high β^* value) against attribute inference than membership inference for a given epoch during training.

1 Introduction

When deploying Machine Learning (ML) models, privacy of training data is a central concern. This encompasses different adversary goals with corresponding threat models. For example, if an attacker could violate our privacy requirements by inferring whether a specific data record was used during training, we would aim to make the model resilient against *membership inference* attacks [22, 24, 26]. On the other hand, if the concern arises from an attacker uncovering sensitive attributes about training data records, we would ensure resilience against *attribute inference* attacks [12, 26]. In practical applications, we are most often concerned with these *specific* privacy threats.

However, the standard approach for ensuring training data privacy is to adopt a much more stringent notion that regards *any* form of information leakage as a privacy violation. Differential privacy (DP) captures this idea by bounding the probability of any record-level information leakage (including membership and attribute inference attacks), regardless of whether this is actually a privacy concern. 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 [10, 13, 20], but few ways of computing bounds against specific privacy attacks [23].

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, age and others, 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 [15, 28]. 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 [11, 19] or numerical errors (e.g., floating point precision) [13]. Further, despite being considered optimal (up to discretization error), accountants generally come with computational costs, which researchers are currently trying to reduce [11].

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 algorithm DP-SGD, and determine simple closed-form (tight) bounds for models trained using this algorithm against these specific 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 error for this approximation, and show that it can be made negligible by tuning the privacy parameters of the algorithm; importantly, for parameters that ensure good privacy, this error becomes very small.

Our theoretical analyses are facilitated by use of the *Bayes security* metric (β^*) [5]. 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., [26]). 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. For example, suppose an algorithm is $\beta^* = 0.98$-secure against MIA; if members and non-members are uniformly distributed, the accuracy of a Bayes-optimal attacker is $0.51 = 1 - {}^{0.98}/_{2}$; The simplicity of our proofs suggests our techniques can be easily extended in the future to other algorithms and privacy threats.

Our paper makes the following contributions:

- We propose a new approach to directly measure privacy of ML models trained using DP-SGD, which addresses the drawbacks outlined above: 1) It is *threat-specific*, in that the metric is directly computed without going via (ε, δ). 2) It streamlines the proof, and makes for a straightforward implementation. Importantly, it is substantially faster to compute compared to modern numerical accountants [10, 13]: it is orders of magnitude faster than state-of-the-art methods for computing the risk against MIA. It also allows us to represent and analyze DP-SGD as used in practice (e.g., using fixed batch sizes).
- We demonstrate that our new approach matches existing techniques in computing tight bounds for membership inference (MIA). while requiring orders of magnitude lesser computation time than prior work.
- 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 train the model for longer and achieve better utility whilst maintaining acceptable privacy, as shown in Figure 1.

The results we present and those in related literature, all assume that a *training-time* attacker who 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 two specific threats: membership inference and attribute inference. In this section, we first provide a brief overview of the DP-SGD algorithm, then formally define the two specific threat models, and finally describe the security metric we use to quantify the risk of a successful attack in each of these threat models.

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 $\{x_1, \ldots, x_N\}$ and a loss function

$\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$, 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 in a (ε,δ)-DP manner as follows: For each step $t = 1,\ldots,T$, sample on average L records from the training set, clip their gradients' norms to be at most 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.

Algorithm 1: DP-SGD($\{x_1,\ldots,x_N\}$, $\mathcal{L}(\theta)$, η_t, σ, L, C)

```
Initialize $\theta_0$ 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 $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$
    Clip gradient
    $\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$
    Add noise
    $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
    Descent
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
end
Output $\theta_T$
```

The privacy parameters (ε,δ) are generally obtained via accounting mechanisms. Along with DP-SGD, Abadi et al. [1] also introduced the Moments Accountant for computing the privacy guarantees for composed mechanisms. Due to the iterative nature of DP-SGD this is essential to obtain accurate privacy guarantees. More recently, Dong et al. [9] introduce f-DP which gives rise to lossless composition. This notion of DP composes all possible (ε,δ) at once and only after the composition converts the privacy guarantee back to a single (ε,δ) pair. However, computing the composition is challenging and several numerical approximations have been developed [13, 18, 21] with currently the Connect-The-Dots algorithm being the fastest [10].

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) membership inference, the attacker aims to ascertain whether a data record appeared in the model's training set. As shown in Game 2, part of the model's training set, D, is sampled from some distribution μ. A further set of $M \geq 2$ *challenge points* is sampled from the same distribution; and from this set, one point x_s^* is sampled according to a *prior* distribution π. The model is trained on $D \cup \{x_s^*\}$, and the intermediate DP-SGD updates $\{\theta_t\}_{t=1}^T$ are revealed to the attacker (who is also assumed to know the game parameters $\mathcal{T},\mu,\pi,N,M,T$). The attacker's goal is to guess *which one* of the challenge points was used for training the model. This game 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. Further, it should be noted that our results still hold if D and the challenge points are sampled from different distributions (e.g., as would be the case for user-level membership inference).

Game 2: MIA-record-level($\mathcal{T},\mu,\pi,N,M,T$)

```
$D \leftarrow \mu^{N-1}$
$x_1^*,\ldots,x_M^* \leftarrow \mu^M$
$s \leftarrow \pi_{\{1,\ldots,M\}}$
$\{\theta_t\}_{t=1}^T \leftarrow \mathcal{T}(D \cup \{x_s^*\})$
$s' \leftarrow \mathcal{A}(\{\theta_t\}_{t=1}^T,\ \{x_i^*\}_{i=1}^M,\ D)$
```

Attribute inference (AI) In attribute inference, the attacker aims to ascertain the value of one or more *sensitive attributes* of a data record, given access to the remainder of that record. As shown in Game 3, a partial training set D and a data record x^* are sampled from some distribution μ; for simplicity, these are chosen to be the same distribution in the game, although our results generalize to arbitrary distributions. Without loss of generality, let a represent the last coordinate of x^*, and let $\phi(x^*)$ be the remainder of the data record. The attacker's goal is to guess a given access to: $\phi(x^*)$, the intermediate models, and the partial training set.

Game 3: AI($\mathcal{T},\mu,\phi,N,T$)

```
$D \leftarrow \mu^{N-1}$
$x^* \leftarrow \mu$
$a \leftarrow$ Last coordinate of $x^*$        // Note: $x^* = (\phi(x^*),\ a)$
$\theta \leftarrow \mathcal{T}(S \cup \{x^*\})$
$a' \leftarrow \mathcal{A}(\{\theta_t\}_{t=1}^T,\ \phi(x^*),\ D)$
```

2.3 Security Metrics

For each threat model, we need to quantify the risk that the attack is successful.

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, 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 $\mathcal{A}(\pi,\theta)$ to indicate an attacker who has access to the model θ (and with prior knowledge π), and $\mathcal{A}(\pi)$ for an attacker who guesses purely based on prior knowledge. By convention, we assume the former is at least as strong as the latter. The advantage 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 a maximal advantage:

$$\begin{aligned}\mathrm{Adv}_\pi &= \frac{Pr[\mathcal{A}(\pi)=S]-Pr[\mathcal{A}(\pi,\theta)=S]}{Pr[\mathcal{A}(\pi)=S]}\\ &= 1-\frac{Pr[\mathcal{A}(\pi,\theta)=S]}{Pr[\mathcal{A}(\pi)=S]}\end{aligned}$$

This is a generalized version of the notion of advantage typically used in the literature. For example, by letting S take a binary value and setting π to be a uniform distribution over the possible values of S, we get $Pr[\mathcal{A}(\pi)=S]={}^1\!/_2$. Substituted into the above, this gives the familiar expression for advantage, as used by e.g., Yeom et al. [26]:

$$\mathrm{Adv} = 2Pr[\mathcal{A}(\pi,\theta)=S]-1$$

Note that this specific notion of advantage cannot be applied to Games 2 and 3 because, in both cases, the secret may have more than two possible values, and need not come from a uniform prior distribution π.

Bayes Security Based on the notion of generalized advantage, we use the *Bayes security metric* proposed by Chatzikokolakis et al. [5], which is defined as:

$$\beta^* = 1-\max_\pi \mathrm{Adv}_\pi \qquad (1)$$

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 [5]). *The generalized advantage is maximized by (equivalently, Bayes security is achieved on) a uniform prior on two secrets:*

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

where u_{s_0,s_1} is a uniform prior on exactly 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:

Proposition 2 (Bayes security and total variation [5]). *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:*

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

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 $\mathrm{tv}(\_,\_)$ *denotes total variation distance.*

Table 1: Summary of notation.

Symbol	Meaning
x^*	Challenge point about which the attacker wishes to learn some property.
$f(x^*)$	Property of interest.
$O_1,\ldots,O_T$	Intermediate model weights output by DP-SGD.
$G_1,\ldots,G_T$	Intermediate (noisy) gradients output by DP-SGD for the challenge point (Equation (2)).
σ	Noise parameter.
C	Gradient norm clipping parameter.
$p={}^L\!/_N$	Poisson sample rate, where N is the size of the training set, and L is a user-chosen parameter.

The relation between β^* and total variation will be an important building block of our formal results.

Proposition 3 (Bayes security and $(0,\delta)$-LDP [5]). *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:*

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

This relation will facilitate the comparison of our security analysis with DP-based analyses.

3 Proof Strategy and Main Result

In this paper, we derive direct proofs for the security of DP-SGD against record-level MIA and AI. These threats, formalized in Game 2 and Game 3, can be unified in the following setup. Let x^* be some data record (i.e., the *challenge point*) about which the attacker aims to infer some property $f(x^*)$. In MIA, the challenge point is chosen from a set of possible challenge points $x^*\in\{x^*_1,\ldots,x^*_M\}$, and $f(x^*)$ is an index to that set such that $x^*=x^*_{f(x^*)}$. In AI, for an arbitrary challenge point x^*, composed of the concatenation $x^*=\phi(x^*)\mid a$, where $a\in\mathcal{A}$ represents a sensitive attribute, the property is $f(x^*)=a$.

Let $S=f(x^*)$ be a random variable representing the secret property. The attacker described in our threat model aims to guess S given the intermediate models output by the DP-SGD algorithm, which we denote by the random vector $O=(O_0,O_1,\ldots,O_T)$. As discussed in preliminaries, we measure the Bayes security, $\beta^*(\textit{Game } 2)$ and $\beta^*(\textit{Game } 3)$, which captures the advantage of an optimal attacker over a naive attacker who guesses S based only on prior information. The relation between S and the outputs is described by the posterior $P_{O|S}$. We therefore need to measure $\beta^*(P_{O|S}) = 1 - \max_\pi \mathrm{Adv}_\pi$, where the advantage is defined for the Bayes optimal adversary who guesses S given information O.

3.1 All You Need Is Two Points ...

In both Games 2 and 3, S can take 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.

To solve this issue, we apply the fact that the generalized advantage is maximized over a uniform prior (Theorem 1). 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 a 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.[1]

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 gradients, rather than intermediate model weights. Formally, the attacker observes output $O = (O_1, \dots, O_T)$ (intermediate model weights), which is defined as:

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

where L is the sampling parameter, N the size of the training dataset, $\mathcal{B}()$ a Bernoulli distribution, and $\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$ is zero-mean isotropic Gaussian noise with variance $\sigma^2 C^2$.

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

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

It is easy to see that $P_{O|S}$ can be obtained from $P_{G|S}$ via post-processing. Since the Bayes security of a channel cannot decrease by postprocessing [5][2] we have:

Corollary 4.

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

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 x^* as O. We shall henceforth study the security of $P_{G|S}$.

3.3 Bayes Security of DP-SGD

Based on Proposition 2, 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 $s_0, s_1 \in \mathbb{S}$. Observe that, because of Poisson sampling, $P_{G|S}$ 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 other DP-SGD analyses (e.g., DP-based) have encountered its analog. Moment 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. [23] uses Monte Carlo estimations, by sampling from the mixture distribution. Both approaches, although valid, come with computational overheads.

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

This section is organized as follows. First, we describe the approximation of a mixture of Gaussians as a single Gaussian. Second, we review a result by Devroye et al. [8] giving a closed-form expression of the total variation between two Gaussian distributions. Finally, we use these results to prove our main statement.

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 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 use the abbreviated notation $p = {}^{L}\!/_{N}$.

Proposition 5. *Let $f_{\mathcal{M}}$ be a Gaussian mixture defined as follows. For a mean vector $\mu = (\mu_1, \dots, \mu_T)$ and covariance*

[1] We observe that, for the special case where π is uniform, a morally equivalent result to the above corollary can be obtained via a reduction from Game 2 to a modified game with $M = 2$. However, this reduction would have a *qualitative* nature of the form “the game for $M = 2$ cannot be harder than the game for $M > 2$”. In contrast, Theorem 1 gives a *quantitative* reduction, which holds for the security metric itself.

[2] This result can also be obtained via the data processing inequality for f-divergences, using the relation between β^* and the total variation distance.

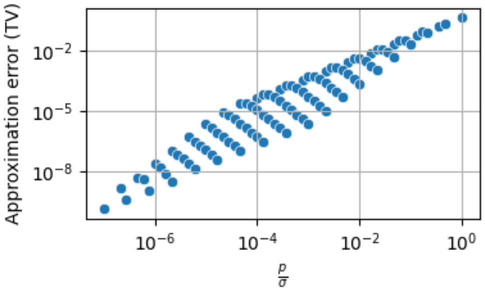


Figure 2: We compare the error induced by approximating a mixture of Gaussians with a Gaussian, as per Proposition 5. The error, measured as the total variation between the original and approximate distributions, is computed via numerical integration. This is done for a fixed $T = 1$. We observe that a small ratio between the sampling rate p and the noise parameter σ ensure the error is negligible for all practical purposes.

matrix $\sigma^s \mu^2 \mathbf{I}_T$, and $C = \max_{j=1}^{T} \mu_j$:

$$f_{\mathcal{M}}(x) = \sum_{b \in \{0,1\}^T} \pi_b f_{\mathcal{N}(\mu b, \sigma^2 C^2)} .$$

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 [7] and Eq. 13 in [14]):

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

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

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

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:

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

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 5 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.

The total variation between two Gaussians For all the threats we consider, as derived in the next sections, determining the Bayes security of DP-SGD reduces to computing the total variation between two Gaussian distributions that are identically-scaled (with isotropic covariation matrix). For this step of the proof, we use the following closed form expression, which was derived by Devroye et al. [8] using a result by Barsov and Ul'yanov [3]:

Corollary 6 (From Barsov and Ul'yanov [3], Theorem 1)**.** *Let $d \geq 1$, $\mu_0, \mu_1 \in \mathbb{R}^d$, $\sigma > 0$. Then,*

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

Main result We can now state our main result, which is based on applying the above observations to the specific case of DP-SGD. The result depends on a parameter, Δ_{s_0, s_1}, that is threat-specific. It is composed of the difference between two *worst-case* gradients at step t, for each step $t \in \{1,...,T\}$; here, the worst case depends on the property f that the attacker wishes to uncover, computed for the two data records that maximize the risk. The value of Δ_{s_0,s_1} will be made explicit for the for the case of MIA (Corollary 8) and AI (Corollary 9) in the next sections.

In the proof of our main theorem, we assume that f is a bijection; that is, there is exactly one challenge point that satisfies $f(x) = s$, for every s. This gives more power to the attacker. This simplification is a technicality used to match $f(x)$ back to its respective challenge point x, and it holds in the cases of membership and attribute inference.

Theorem 7. *The Bayes security of DP-SGD with respect to the property guessing threat described in Section 3 is:*

$$\beta^*(P_{O|S}) \geq 1 - \mathrm{erf}\left(p\frac{\max_{s_0,s_1\in\mathrm{dom}(S)} \|\Delta_{s_0,s_1}\|}{2\sqrt{2}\sigma C}\right) + O\left(\frac{\sqrt{pT}}{\sigma}\right),$$

where $\|\cdot\|$ is the Frobenius norm, $\bar{\mathbf{g}}(x^*) = (\bar{\mathbf{g}}_1(x^*),...,\bar{\mathbf{g}}_T(x^*))$, and $\Delta_{s_0,s_1} = \bar{\mathbf{g}}(f^{-1}(s_0)) - \bar{\mathbf{g}}(f^{-1}(s_1))$.

Proof. By Corollary 4, the relation between Bayes security and the total variation Proposition 2, and Theorem 1:

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

We now determine the total variation term. $P_{G|S}$ is a Gaussian mixture: each of its T components is a Gaussian probability distribution centered in either $\bar{\mathbf{g}}(f^{-1}(s_0))$ (with probability p) or 0 (otherwise).

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 5.

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

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 6. □

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 7 provides a bound for the Bayes security of DP-SGD against a fairly generic attack, where the attacker is tasked with guessing a property f of a challenge point. In this section, we specify this property to be the *membership* of the training point.

The Bayes security of DP-SGD against MIA follows as a corollary to Theorem 7:

Corollary 8. *The Bayes security of DP-SGD against record-level MIA is:*

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

Proof. Using the notation of Theorem 7:

$$\begin{aligned}\max_{s_0,s_1\in\mathrm{dom}(S)} \|\Delta_{s_0,s_1}\| &= \max_{x_0^*,x_1^*\in D} \|\bar{\mathbf{g}}(x_0^*) - \bar{\mathbf{g}}(x_1^*)\| \\ &\leq 2C\sqrt{T}.\end{aligned}$$

Applying Theorem 7 concludes the proof. □

DP-SGD parameters selection. We observe that Theorem 7 makes it possible to *cheaply* decide on which parameters to select before running DP-SGD, given a desired level of MIA-resilience. Suppose, for example, that an application requires $\beta^* \geq 0.98$; this corresponds, assuming a uniform prior between members and non-members, to a 50% probability of the success for the attacker. Further, suppose we wish to train for $T = 5k$ steps. We can select the noise parameter σ and the sampling rate p based on the relation:

$$p = \frac{\mathrm{erf}^{-1}(1-\beta^*)\sqrt{2}}{\sqrt{T}}\sigma; \qquad (3)$$

in this example, we have $p \approx 0.00035\sigma$. By using this relation, not only will we be guaranteed the desired level of protection; we will also get the *tightest* values for which it is satisfied.

Figure 3 shows how to pick noise and sampling rate to meet target levels of security against MIA. Figure 4 shows how security varies w.r.t. the number of steps T for which DP-SGD is run.

4.1 Bayes Security and Differential Privacy

Thanks to its long and rich research history, various methods have been developed that satisfy the (approximate) DP definition. Given a mechanism that satisfies (ε,δ)-DP, one can generally match back its privacy parameters to the threat model of interest. By design, DP targets the strongest possible

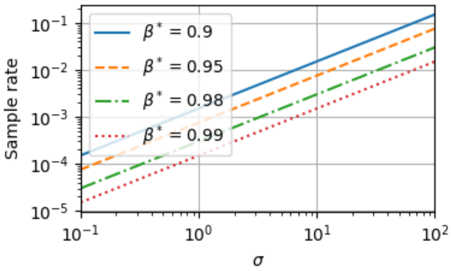


Figure 3: Bayes security against MIA: picking the noise and sampling rate to achieve a desired level of security.

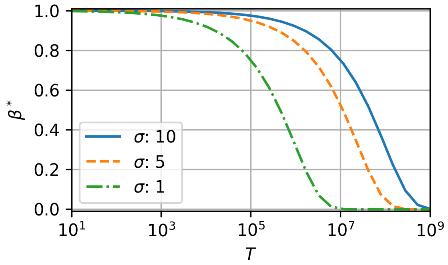


Figure 4: Bayes security against MIA. $N = 100k$, $L = 10$. Dashed line is the approximate bound.

attacker possible: the membership inference attacker. This favors its comparison with our MIA-specific results. In this section, we i) compare state-of-the-art DP estimators, adapting them to be β^* estimators, ii) relate (ε, δ) to the notion of β^* (and advantage), and iii) observe how our results could be used in the future as a cheap (ε, δ) estimator.

4.1.1 Comparison with PRV Accountant

Thanks to the relation with $(0, \delta)$-DP (Proposition 3), it is possible to compare our security bounds with equivalent ones estimated via the DP accountant method. In particular, we can use the PRV accountant to estimate the value δ, with the constraint $\varepsilon = 0$, and compare this with $1 - \beta^*$.

Figure 5 shows this comparison. We observe that the two estimates look identical, on a variety of parameter sets. On the one hand, this validates our new approach to analyzing DP-SGD. On the other hand, considering that PRV is optimal (up to discretization error), it suggests that our bounds are also optimal for the considered threat model.

4.1.2 Comparison with (ε, δ)-DP

Bayes security and $(0, \delta)$-DP are related by the equality $\beta^* = 1 - \delta$. One may wonder how DP-SGD behaves for values of $\varepsilon > 0$.

We make this comparison by exploiting a bound by Humphries et al. [15], relating the advantage of an MIA attacker with the (ε, δ) parameters of the privacy mechanism:

$$\mathrm{Adv} \leq \frac{e^{\varepsilon} - 1 + 2\delta}{e^{\varepsilon} + 1} . \tag{4}$$

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 PRV accountant for DP-SGD, and then plugging them into Equation (4). In this plot, a direct comparison with Bayes security is enabled by the equivalence $\beta^* = 1 - \mathrm{Adv}$; β^* is here computed via Theorem 7.

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.

4.1.3 Corollary 8 as a (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 8 to obtain a rough estimate of (ε, δ)-DP. To this end, we can once again exploit Equation (4) by Humphries et al. [15].

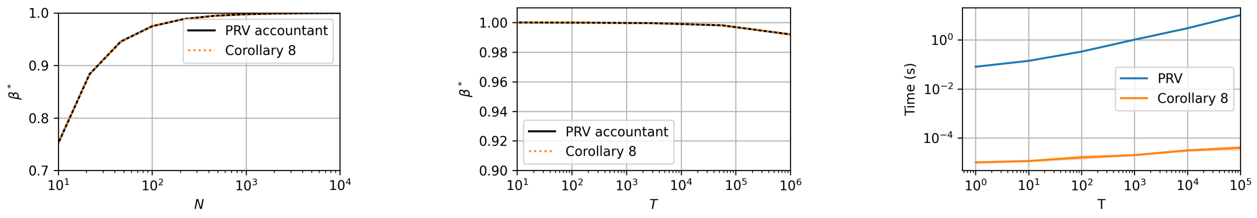
By the correspondence between the advantage and Bayes security ($\mathrm{Adv} = 1 - \beta^*$), we obtain a bound on ε as follows. Let β^* be the Bayes security of DP-SGD computed as per Corollary 8; then for any choice of $\delta \in [0, 1)$, we get:

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

This bound can serve as a cheap alternative to more computationally expensive methods (e.g. numerical accountants) for estimating ε. Two aspects should be remarked. First, the above inequality is highly susceptible to under-estimations of β^*; therefore, it is important that a good estimate is used. Second, it should be noted that this bound is necessarily loose. The inequality by Humphries et al. [15], whilst tight, applies to *any* (ε, δ)-DP algorithm: therefore, it is possible to improve on their inequality, and, therefore, the above estimate, for the special case of DP-SGD; this is what more advanced (ε, δ)-DP estimators do.

5 Attribute Inference

In this section, we apply Theorem 7 to prove the security of DP-SGD against AI. First, we discuss the limits of *any* security analysis: without making assumptions, one cannot improve on the MIA bound (Corollary 8). We mitigate this issue by providing data-dependent bounds for AI, and providing a modification of DP-SGD to compute them. Second, we study whether a data-dependent security analysis has any security implications. Finally, we discuss the computational



(a) $T = 10$

(b) $N = 100k$

(c) $N = 100k$.

Figure 5: Bayes security of DP-SGD against record-level MIA. The Bayes security is compared with the equivalent metric obtained via a DP accountant. The bottom plot compares the running times of the two. $L = 10$, $C = 1$, $\sigma = 10$.

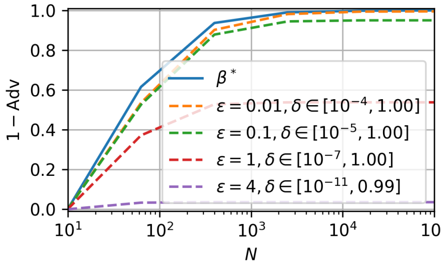


Figure 6: (ε, δ)-DP of DP-SGD. We set $N = 100k$, $L = 10$, $C = 1$, $T = 1$. Here, β^* is computed via Corollary 8.

time overheads of this method, and we discuss ways to improve it.

5.1 Limits of *any* AI Analysis

Before discussing our main bound for AI, it is important to understand what is achievable by a security analysis of DP-SGD under this threat model. 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 (Corollary 8) 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 a_S given partial information $\phi(x^*)$ 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 $\phi(x^*)$ and the clipped and noisy gradient: $\bar{\mathbf{g}}_t(x^*) + \text{Noise}$; 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 create a contrived gradient function, which returns $\bar{\mathbf{g}}_t(\phi(x^*) \mid a_0) = -\bar{\mathbf{g}}_t(\phi(x^*) \mid a_1) = (C, 0, \ldots, 0)$, 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 8, 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 $\|\bar{\mathbf{g}}_t(\phi(x) \mid a_i) - \bar{\mathbf{g}}_t(\phi(x) \mid a_j)\|$, for all training points x and all possible attribute values a_i, a_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 our computational solution 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 8) to the case of AI. We do this by specializing the definition of Δ_{s_0,s_1}, as defined in our main result, to the case of AI. Under the notation we introduced in Section 3, an AI attacker given partial information $\phi(x^*)$ about a data record x^* aims to guess the property $f(x^*)$, such that $x^* = \phi(x^*) \mid f()x^*)$.

Corollary 9. *The Bayes security of DP-SGD against AI is:*

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

where $R = (R_1, \ldots, R_T)$ with

$$R_t = \max_{x^* \in L_t} \max_{a_0, a_1 \in \mathcal{A}} \|\bar{\mathbf{g}}_t((\phi(x^*), a_0)) - \bar{\mathbf{g}}_t((\phi(x^*), a_1))\|,$$

where $L_t \subseteq D$ is the batch sampled at step t.

Proof. We bound $\max_{s_0,s_1 \in \text{dom}(S)} \|\Delta_{s_0,s_1}\|$ as defined in Theorem 7.

$$\begin{aligned}\max_{s_0,s_1 \in \text{dom}(S)} \|\Delta_{s_0,s_1}\| & \\ &\leq \max_{a_0,a_1 \in \mathcal{A}, x^* \in D} \|\bar{\mathbf{g}}(\phi(x^*) \mid a_0) - \bar{\mathbf{g}}(\phi(x^*) \mid a_1)\| \\ &\leq \sum_{t=1}^{T} \max_{a_0,a_1 \in \mathcal{A}, x^* \in D} \|\bar{\mathbf{g}}_t(\phi(x^*) \mid a_0) - \bar{\mathbf{g}}_t(\phi(x^*) \mid a_1)\|\end{aligned}$$

In the theorem statement, $x^* \in L_t$, where L_t is the batch sampled at time t. This is allowed by observing that, if x^* is not included in the batch at step t, it cannot influence the model weights (and gradients) at that step. □

Algorithm 4: AI-resilient-SGD($\{x_1,\ldots,x_N\}, \mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i), \eta_t, \sigma, L, C, \mathcal{A}$)

```
Initialize $\theta_0$ 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 $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$
    Compute gradient bound w.r.t. attribute's value
    $R_t = \max_{x^* \in L_t} \max_{a_0,a_1 \in \mathcal{A}} \|\bar{\mathbf{g}}_t((\phi(x^*),a_0)) - \bar{\mathbf{g}}_t((\phi(x^*),a_1))\|$
    Clip gradient
    $\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i)/\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$
    Add noise
    $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
    Descent
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
end
Output $\theta_T$
```

We describe how DP-SGD can be adapted to compute the values of R_t exactly, for each step. These values are then plugged in Corollary 9 to compute its Bayes security against AI. 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 instruments DP-SGD for calculating R_t exactly. For every batch L_t and for every point $x^* \in L_t$ in the batch, this algorithm augments $\phi(x^*)$ with all the valid completions $a \in \mathcal{A}$, and determines the maximum distance between the two. Security for that step is determined by plugging the vector $R = (R_1, \ldots, R_T)$ in Corollary 9.

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, the metric itself, 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, given access to this metric?

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 x_b^* or attribute value a, depending on the threat model).

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, x_0^* and x_1^*, are both such that $R_t = \max_{a_0,a_1 \in \mathcal{A}} \|\bar{\mathbf{g}}_t((\phi(x_b^*),a_0)) - \bar{\mathbf{g}}_t((\phi(x_b^*),a_1))\|$; the parameter R_t^b, computed for either challenge point x_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 for deployment, the security parameter estimated for the AI analysis is not revealed to the public. Alternatively, one can simply reveal that β^* is above some independently chosen threshold. One may also consider adding noise to R_t to prevent this potential issue; clearly, however, this strategy would deteriorate the estimated security analysis.

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, and, therefore, β^*, is the same regardless of the value of the sensitive attribute a. 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 5.5), we observe that the time overhead is acceptable for small attribute spaces. Nevertheless, as $\mathcal{A}$ grows, this cost becomes unfeasible. 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 $a_0, a_1 \in \mathcal{A}$ 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.[3] In this case, we can safely expect that the two values maximizing the risk for the attribute will be the two extremes, $\{0,1\}$. This observation can reduce the computation of Algorithm 4 to a manageable cost.

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,

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

which is defined as follows. Let (M,d) be a metric space, where M is a finite set, and d is some metric. 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 [5, 16, 25]. In this paper, we consider a simple $O(N)$ solution, where $N = |M|$, which gives a lower bound based on the triangle inequality: for any choice $x \in M$, $\text{diam}_M \leq 2 \max_{y \in M} d(x,y)$.

To estimate R_t, we pick the mean vector of the gradients, $m \in \{\bar{\mathbf{g}}_t((\phi(x^*),a))\}_{a \in \mathcal{A}}$, and we estimate R_t as:

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

The choice of a *lower* bound here is security-motivated: it describes the worst-case for the victim.

Note that this estimate can be improved either by picking more carefully the point m, or by running this algorithm for various choices of m and then choosing the one giving the tightest bound. Despite the approximation given by the triangle inequality, in our experiments we observed that this approximation is close enough to the real value. Nevertheless, practical applications may consider solutions that give tighter bounds (e.g., [16]).

Future work 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.

5.5 Empirical Evaluation

We evaluate our analyses on two datasets. In these experiments, we wish to: i) evaluate the practical feasibility of our analyses in computational terms, and ii) measure the benefits of the two analyses.

Datasets We select two tabular datasets; this makes it meaningful to conduct an attribute inference analysis. First, we select the Adult Census Income dataset (`Adult`), which is composed of 32561 records with 108 attributes each (after one-hot-encoding). The `Adult` dataset 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), 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 $\{17, 18, ..., 90\}$.

Second, we consider the Purchase dataset (`Purchase100`). This dataset has 197324 records and 600 attributes. Each record correspond to one customer, and each (binary) attribute indicates whether the customer bought a particular item. Because of its size, this dataset enables evaluating i) how well the analyses scale to larger datasets, and ii) what are the privacy-utility gains given by our analyses. For the AI analysis, we select the first attribute (purchase) as the sensitive one.

Models and setup We train two fully connected neural networks as described by Bao et al. [2], implemented via `pytorch`. We instrumented `Opacus` [27] to support our AI analysis, as a callback function that is run at every step. *Upon publication of this manuscript, we will share our code and, and provide a version that can be integrated in Opacus.*

Privacy parameters We use Equation (3) 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 suboptimal security against MIA enables observing the advantage in 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 `Purchase100` dataset. This, paired with the training set size N, enables determining the following noise parameters are required: $\sigma = 3.51$ (`Adult`) and $\sigma = 1.8$ (`Purchase100`).

5.5.1 Running Time

First, we evaluate the computational costs of our analysis. We present the measurements for the `Adult` dataset, which is the harder dataset 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 7 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 analysis (Section 4.1.1) 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 5.5.3, we observe that this may be a fair price to pay given their advantages from a privacy-utility perspective.

Finally, we observe that the approximate analysis reduces the costs by one order of magnitude. In the next part, we

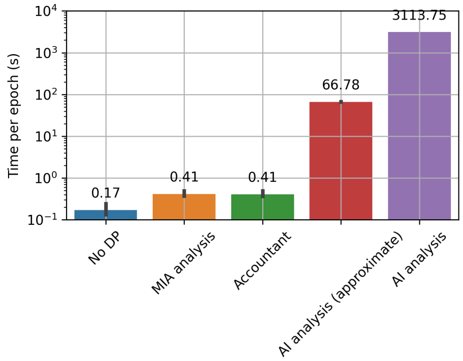


Figure 7: Running time comparison: average time per epoch, measured across 10 epochs. Adult dataset (sensitive attribute has 73 possible values), 26 steps per epoch.

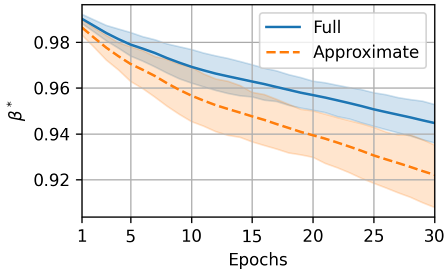


Figure 8: Bayes security on the Adult dataset, estimated via the full and approximate AI analysis.

evaluate the privacy loss given by this approximate analysis.

5.5.2 AI: Full vs Approximate Analysis

In the previous part, we observed that the approximate AI analysis heavily reduces the computational costs. We observe that the approximate analysis can never indicate more security than what there actually is: it is a lower bound of the full AI analysis by construction.

Figure 8 compares the Bayes security estimates of the two analyses, for an increasing number of epochs. 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. 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 30 epochs (780 steps) are $\beta^* = 0.992$ for the full analysis and $\beta^* = 0.990$ 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.

5.5.3 Bayes Security of DP-SGD for the Adult and Purchase100 Datasets

We now compare the estimates of the MIA and AI analyses. As a reference, we include a $(\varepsilon, \delta = 10^{-5})$-DP estimate provided by a numerical accountant. Figure 1 shows the results respectively for the Adult and Purchase100 dataset. When comparing the two β^*-guided analyses, it is important to keep in mind that MIA represents the worst-case scenario for AI. In other words, it prevents AI against *any* attribute; the sensitive attribute could be the entire data record, for example. We observe that, if the sensitive attribute is known in advance, then the AI analysis can give remarkably tighter bounds on its security. By exploiting the (ε, δ)-DP relation, we observe that a very large value of ε is sufficient to prevent the AI attack.

On the other hand, we observe that, for the chosen parameters, the risk of MIA is high even after training for few epochs. Remarkably, this low security matches a value ε that would generally be considered to be safe. This enforces the fact that an analysis based on the attacker advantage (or, equivalently, Bayes security) for *specific* threat models gives a better understanding of the privacy risks.

6 Towards Inference-time Attackers

Training-time attacker Both DP-inspired methods 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 PRV accountant or the approach by Humphries et al. [15]. 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 (i.e., higher) lower 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 x^*, (possibly) sampled at step t, has on the subsequent gradient functions $\bar{\mathbf{g}}_{t+1}(\cdot), \bar{\mathbf{g}}_{t+2}(\cdot), \ldots$. Let $t > 1$. In Section 3.2, we could disregard the effect of $\bar{\mathbf{g}}_t(x)$ for points $x \neq x^*$. The reason is that $\bar{\mathbf{g}}_t(x)$ did not bear any *additional* information about the challenge point to the adversary; indeed they had access to $\bar{\mathbf{g}}_{t-1}(x)$ as well. However, for an inference-time attacker, this reasoning does not apply. Since $\bar{\mathbf{g}}_t(\cdot)$ might have seen the effect of x^* in a previous step, and because the attacker is not given access to $\bar{\mathbf{g}}_{t-1}(\cdot)$, 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 x^* is sampled at step t. Then we can write a combinatorial expression that factors the likelihood that gradients (at points $x \neq x^*$) were tainted by the presence of x^*. 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 "x^* was sampled before step t" grows exponentially with t; leading to trivial bounds.

Overall, 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 may be a good strategy to attack this problem; however, it should be noted, this would require making assumptions about the gradient function itself.

7 Related Work

Mahloujifar et al. [23] In concurrent work, Mahloujifar et al. [23] 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.

Importantly, in Section 4.1.2 we show that, in the specific case of membership inference, one cannot improve on estimates obtained via state of the art (ε, δ)-DP estimators. This means that both the analysis by Mahloujifar et al. [23], and the special case of our main result to membership inference, cannot give a tighter security analysis. Yet, we observe that our method has other benefits; e.g., it is a closed form expression which can be computed in negligible time.

Alternative Metrics Carlini et al. [4] suggest that accuracy-based metrics do not appropriately capture the risk of membership inference attacks. They recommended instead 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. In contrast, we use Bayes security, a worst-case metric that captures the risk for the two data records for which determining membership is the easiest. We measure risk against a Bayes optimal attacker [5], and bound membership inference advantage, the difference between the true positive and false positive rate of an attack. This implies (loose) bounds on TPR@FPR metrics.

For attribute inference, our metric corresponds to the *alternative attribute advantage* defined by Yeom et al. [26, Definition 6]. They argue that this definition measures the inference risk due to a model generalizing well to the training data distribution or to it overfitting its training data, while a definition of advantage analogous to the one used for membership inference would measure only the risk due to overfitting.

8 Conclusion and Future Work

We present a new analysis strategy for studying the security of DP-SGD for specific threats. Using this, we derived a new closed-form tight bound for the risk against MIA, which is orders of magnitude faster to compute than previous approaches. We also present a data-dependent bound on the risk against AI. Our empirical results show that DP-SGD can be significantly more resilient against AI than MIA, which ultimately leads to better utility in applications where MIA is not a concern.

Opportunities for future work include further improvements to the computational efficiency of our AI bounds analysis. When using the approximate algorithm described in Section 5, 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) [6, 17] 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. Additionally, future work may be able to adapt our analysis for proving security against weaker (albeit realistic) inference-time attackers who only observe the final model. Finally, 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 analysis 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] 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.

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

[4] 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.

[5] Konstantinos Chatzikokolakis, Giovanni Cherubin, Catuscia Palamidessi, and Carmela Troncoso. The bayes security measure. *arXiv preprint arXiv:2011.03396*, 2020.

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

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

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

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

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

[19] Antti Koskela and Antti Honkela. Computing differential privacy guarantees for heterogeneous compositions using fft. *arXiv preprint arXiv:2102.12412*, 2021.

[20] Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In *23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*, pages 2560–2569. PMLR, 2020. URL https://proceedings.mlr.press/v108/koskela20b.html.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.