“What do you want from theory alone?” Experimenting with Tight Auditing of Differentially Private Synthetic Data Generation

Meenatchi Sundaram Muthu Selva Annamalai, *University College London;* Georgi Ganev, *University College London and Hazy;* Emiliano De Cristofaro, *University of California, Riverside*

https://www.usenix.org/conference/usenixsecurity24/presentation/annamalai-theory

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

“What do you want from theory alone[*]?” Experimenting with Tight Auditing of Differentially Private Synthetic Data Generation[†]

Meenatchi Sundaram Muthu Selva Annamalai[1], Georgi Ganev[1,2], Emiliano De Cristofaro[3]

[1] *University College London* [2] *Hazy* [3] *University of California, Riverside*

Abstract

Differentially private synthetic data generation (DP-SDG) algorithms are used to release datasets that are structurally and statistically similar to sensitive data while providing formal bounds on the information they leak. However, bugs in algorithms and implementations may cause the actual information leakage to be higher. This prompts the need to verify whether the theoretical guarantees of state-of-the-art DP-SDG implementations also hold in practice. We do so via a rigorous *auditing* process: we compute the information leakage via an adversary playing a distinguishing game and running membership inference attacks (MIAs). If the leakage observed empirically is higher than the theoretical bounds, we identify a DP violation; if it is non-negligibly lower, the audit is loose.

We audit six DP-SDG implementations using different datasets and threat models and find that black-box MIAs commonly used against DP-SDGs are severely limited in power, yielding remarkably loose empirical privacy estimates. We then consider MIAs in stronger threat models, i.e., passive and active white-box, using both existing and newly proposed attacks. Overall, we find that, currently, we do not only need white-box MIAs but also worst-case datasets to *tightly* estimate the privacy leakage from DP-SDGs. Finally, we show that our automated auditing procedure finds both known DP violations (in 4 out of the 6 implementations) as well as a new one in the DPWGAN implementation that was successfully submitted to the NIST DP Synthetic Data Challenge.

The source code needed to reproduce our experiments is available from https://github.com/spalabucr/synth-audit.

1 Introduction

In modern data-driven ecosystems, organizations are often compelled or willing to share data within and with each other [14]. However, even if it is “sanitized,” “anonymized,” or aggregated, sharing data can still lead to severely violating the privacy of the data subjects [10]. In this context, Synthetic Data Generation (SDG) algorithms have been proposed as a potential mitigation; by learning the underlying distribution of the sensitive data and then sampling “fresh” synthetic data points from it, SDGs enable entities to generate and release artificial data that, ostensibly, only statistically resembles the real data. However, without formal privacy protections, SDGs can easily leak sensitive user data [6,26,28,60,70].

Method (Implementation)	**Threat Model**	**Violation**
PrivBayes (DS)	Black-box	Metadata
PrivBayes (DS)	White-box	Pre-processing
PrivBayes (Hazy)	Black-box	Metadata
MST (Smartnoise)	Black-box	Metadata
DPWGAN (NIST)	Active White-box	Early stopping
DPWGAN (Synthcity)	Black-box	Metadata
DPWGAN (Synthcity)	Black-box	PRNG Reuse

Table 1: Overview of identified privacy violations.

The standard, rigorous way to define algorithms with formally bounded information leakage is through Differential Privacy (DP) [19]. Researchers have proposed a number of SDGs that satisfy DP, aka DP-SDGs [8,11,32,37,43,69,75]. Particularly in the tabular data domain [36,43,75], DP-SDGs have started to see real-world adoption; e.g., in 2021, Microsoft and the UN International Organization for Migration released a (DP) synthetic dataset that describes victim-perpetrator relations in the context of human trafficking.

However, bugs that lead to DP violations have been found in several popular DP tools [9,38,49], including in tabular DP-SDGs [38,60]. This motivates the need to *audit* state-of-the-art DP-SDG implementations, i.e., designing and executing experiments to derive *empirical* privacy leakage estimates. These are then compared against the *theoretical* (provable) DP guarantees to verify the correctness of implementations [38] and/or detect DP violations [29,30,49].

The auditing process often relies on membership inference attacks (MIAs), where an adversary attempts to learn whether or not a given record was used as input to the algo-

[*] See https://www.imdb.com/title/tt15398776/quotes/?item=qt6969434.

rithm, following a distinguishing game meant to mirror the DP definition [60]. The threat models in which MIAs can be mounted include what, in Section 3.2, we define as black- and white-box settings; in the former, the adversary only has access to the synthetic data, while, in the latter, she also sees the trained generative model and its internal parameters.

Prior work studying privacy in DP-SDGs has mostly focused on *black-box* attacks and used randomly sampled (*average-case*) training datasets [60]. Alas, these are likely to yield *loose* empirical bounds [28], i.e., the empirical estimates are not close to the theoretical DP guarantees. Conversely, prior work auditing DP algorithms in discriminative settings has obtained tight estimates by considering *active* white-box [50] attacks, where the adversary can manipulate the training process by inserting arbitrary canary gradients [49, 51].

In this work, we present a comprehensive audit of DP-SDGs. This prompts two main research questions: *1)* How *tightly* can we empirically estimate privacy leakage in DP-SDGs? *2)* How do different threat models and datasets affect tightness? To this end, we design an experimental framework including MIAs against different DP-SDG implementations, training datasets, and threat models. Using our framework, we audit three state-of-the-art tabular DP-SDG algorithms (PrivBayes [75], MST [43], and DPWGAN [4]), considering two independent implementations for each algorithm.

Main Findings. Our analysis shows that:

- Common black-box MIAs like the distance to closest record (DCR) heuristic are ineffective at exploiting privacy leakage from DP-SDGs.

- White-box and active white-box attacks provide much tighter empirical privacy estimation, especially with specially crafted *worst-case* datasets. For instance, for MST at theoretical $\varepsilon = 4.0$, white-box auditing produces empirical privacy estimation of $\varepsilon_{emp} = 3.10$ compared to black-box's meaningless estimates ($\varepsilon_{emp} = 0.00$).

- The tightest possible settings may be implementation-dependent, i.e., we need different *worst-case* datasets and threat models to achieve tight empirical privacy estimates for different DP-SDG implementations. E.g., *passive* white-box audits of PrivBayes and MST are tight, while DPWGAN requires *active* white-box attacks.

- As summarized in Table 1, we find DP violations in four out of the six implementations we study, due to learning metadata directly from the input. We also identify a new DP violation in the DPWGAN implementation successfully submitted to the NIST DP Synthetic Data Challenge [54].

Contributions. In summary, our main contributions include:

1. We perform the first large-scale audit of DP-SDG algorithms and their implementations.
2. We craft implementation-specific worst-case datasets for DP-SDGs, which enables us to achieve *tight* audits.
3. We present the first white-box MIAs against PrivBayes and MST.

2 Preliminaries

We now introduce the concepts of differential privacy, auditing, membership inference, and synthetic data generation.

2.1 Differential Privacy (DP)

Definition 1 (Differential Privacy (DP) [19])**.** A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is (ε, δ)-differentially private if for any two neighboring datasets $D, D' \in \mathcal{D}$ and $S \subseteq \mathcal{R}$, it holds:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta$$

Definition 1 describes the so-called *approximate* DP variant, which is a relaxation of the original ("*pure*") DP definition whereby $\delta = 0$. We also consider two variants of DP depending on the definition of *neighboring datasets*: 1) *add/remove*, aka unbounded, and 2) *edit*, aka bounded, DP. The former corresponds to inserting/deleting a single record from the dataset ($|D| = |D'| \pm 1$); the latter entails replacing a single record with another ($|D| = |D'|$).

An important property of DP is given by the *post-processing* theorem, which lets us use the output of DP mechanisms freely without worrying about further privacy leakage.

Theorem 1 (Post-Processing)**.** Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be an (ε, δ)-DP mechanism and $f : \mathcal{R} \to \mathcal{R}'$. Then $f \circ \mathcal{M} : \mathcal{D} \to \mathcal{R}'$ also satisfies (ε, δ)-DP.

2.2 Auditing DP

Implicit to the DP definition is a theoretical limit on any adversary's ability to distinguish between outputs of an (ε, δ)-DP mechanism $\mathcal{M}$ (i.e., $\mathcal{M}(D)$ and $\mathcal{M}(D')$). When observed in practice, this limit can be used to estimate the empirical guarantees provided by a DP mechanism. The process of *auditing* DP entails verifying the *theoretical* guarantees provided by $\mathcal{M}$ by running an experiment where an adversary attempts to distinguish between $\mathcal{M}(D)$ and $\mathcal{M}(D')$ and estimating the *empirical* guarantees $(\varepsilon_{emp}, \delta)$ from the adversary's success.

Informally, when auditing a DP mechanism $\mathcal{M}$, $\mathcal{M}$ is repeatedly run on a pair of fixed neighboring datasets D and D' to generate two sets of observations $O = \{o_1, o_2, \ldots\}$ and $O' = \{o'_1, o'_2, \ldots\}$, respectively. Next, an adversary attempts to distinguish between the two sets of outputs, which results in a false positive rate α and a false negative rate β. (We provide a formal definition in Section 3.3). Then, upper bounds $\overline{\alpha}$ and $\overline{\beta}$ can be calculated using Clopper-Pearson confidence intervals, as done in previous work [49, 51]. Finally, the upper bounds on α and β are converted back into an empirical lower bound

ε_{emp} using two known methods, i.e., auditing using either the (ε,δ)-DP or the μ-GDP definition, described below.

Maximum auditable ε. The empirical lower bound, ε_{emp}, has a confidence level that follows the upper bounds' confidence level. At the same time, this imposes an inherent *limit* on the maximum ε_{emp} that can be derived in this way, which we refer to as the *maximum auditable* ε. Intuitively, even if the adversary can perfectly distinguish between O and O' (i.e., $\alpha = \beta = 0$), $\overline{\alpha}$ and $\overline{\beta}$ are lower bounded by the confidence interval thus resulting in an upper bound on ε_{emp}.

Auditing using (ε,δ)-DP In general, any mechanism that satisfies (ε,δ)-DP bounds the possible false positive rates (α) and false negative rates (β) attainable by an adversary to the following privacy region [33]:

$$\mathcal{R}(\varepsilon,\delta) = \{(\alpha,\beta) | \alpha + e^{\varepsilon}\beta \geq 1-\delta \wedge e^{\varepsilon}\alpha + \beta \geq 1-\delta \wedge \\ \alpha + e^{\varepsilon}\beta \leq e^{\varepsilon}+\delta \wedge e^{\varepsilon}\alpha + \beta \leq e^{\varepsilon}+\delta\}$$

An empirical lower bound ε_{emp} can be calculated according to the privacy region using the following equation:

$$\varepsilon_{emp} = \max\left\{\ln\left(\frac{1-\overline{\alpha}-\delta}{\overline{\beta}}\right), \ln\left(\frac{1-\overline{\beta}-\delta}{\overline{\alpha}}\right), 0\right\} \quad (1)$$

When auditing pure DP, we can use Eq. 1 with $\delta = 0$.

Auditing using μ-GDP. While the (ε,δ)-DP auditing method applies in *general* to all approximate DP mechanisms, Nasr et al. [49] note that mechanisms can have privacy regions that are *specific* to the mechanism as well. For example, mechanisms that satisfy μ-Gaussian Differential Privacy (GDP) also satisfy approximate DP but define a much smaller subset of $\mathcal{R}(\varepsilon,\delta)$ as its privacy region. Therefore, we can audit μ-GDP mechanisms by first converting the bounds on α and β into a lower bound on μ using the following equation:

$$\mu_{emp} = \Phi^{-1}(1-\overline{\alpha}) - \Phi^{-1}(\overline{\beta}) \quad (2)$$

We can then convert the lower bound μ_{emp} to a lower bound ε_{emp} using the following theorem for a fixed δ:

Theorem 2 (μ-GDP to (ε,δ)-DP conversion [18])**.** A mechanism is μ-GDP iff it is $(\varepsilon,\delta(\varepsilon))$-DP for all $\varepsilon \geq 0$, where:

$$\delta(\varepsilon) = \Phi\left(-\frac{\varepsilon}{\mu}+\frac{\mu}{2}\right) - e^{\varepsilon}\Phi\left(-\frac{\varepsilon}{\mu}-\frac{\mu}{2}\right) \quad (3)$$

2.3 Membership Inference Attacks (MIAs)

In a membership inference attack (MIA), the adversary aims to determine if a target record, x_T, was used in input to a function – e.g., aggregation [57], training a model [59], etc. In recent years, a number of MIAs against machine learning models have been presented that consider various threat models and settings [12, 26, 59, 72].

In the DP auditing setting, we define an MIA as a function that takes in input the two neighboring datasets (D, D'), the target record (x_T), the mechanism being audited ($\mathcal{M}$), and a single output y of the mechanism run on D or D'. The MIA function outputs a (possibly unbounded) *score*, s, that represents the confidence the attack assigns to the event that D was the input to the mechanism based on the output (i.e., $y \sim \mathcal{M}(D)$). In short, we define the MIA function as:

$$s \leftarrow \text{MIA}(x_T, y; \mathcal{M}, D, D'). \quad (4)$$

2.4 Synthetic Data Generation (SDG)

Synthetic data generation (SDG) algorithms take in input an *original* dataset D and output a *synthetic* dataset S. Typically, a generative model $\mathcal{G}$ is first fit on D using a (possibly randomized) fitting function, i.e. $\mathcal{G} \sim \text{GM}(D)$. A synthetic dataset with m records is then sampled from this model, i.e., $S \sim \mathcal{G}^m$.

In differentially private synthetic data generation algorithms (DP-SDGs), the fitting function GM itself typically satisfies DP. That is, the probability that the adversary can infer if a given generative model $\mathcal{G}$ was fit on D or D', i.e., $\mathcal{G} \sim \text{GM}(D)$ or $\mathcal{G} \sim \text{GM}(D')$, is bounded by the ε parameter. The guarantees of the overall SDG algorithm then follow from the post-processing theorem, as the synthetic dataset is simply sampled from the fitted generative model. However, DP-SDGs might pre-process the dataset without proper DP accounting, which in practice can result in DP violations [60].

3 Auditing DP-SDG Algorithms

3.1 Overview

We now set out to audit a differentially private synthetic data generation (DP-SDG) algorithm $\mathcal{M}_{\text{SDG}}$ using an adversary and a distinguishing game; given neighboring datasets D and D', the adversary distinguishes between outputs from $\mathcal{M}_{\text{SDG}}(D)$ and $\mathcal{M}_{\text{SDG}}(D')$. More precisely, she distinguishes using a membership inference attack (MIA). The attack's success rate – i.e., the number of false positives and false negatives – is then used to compute a lower-bound empirical estimate, ε_{emp}, of the privacy leakage.

We consider the auditing procedure to be "tight" if the empirical estimate ε_{emp} is close to the theoretical guarantee ε. Thus, the auditing procedure can be used to identify privacy violations [16, 38, 49], if $\varepsilon_{emp} \gg \varepsilon$. It can also be used to determine if the theoretical guarantees are loose or if there is significant room for the membership inference attacks to be improved [51], if $\varepsilon_{emp} \ll \varepsilon$.

We instantiate a range of MIAs with varying adversarial capabilities and study the impact of auditing decisions on tightness. In the rest of this section, we define the threat models considered in this work, formalize the DP distinguishing game, and define and introduce the methodology used to select *worst-case* target records and neighboring datasets.

3.2 Threat Models

Our analysis considers progressively stronger threat models (introduced below) to study the power of the adversary needed to achieve tight empirical estimates. While there have been many different definitions of threat models introduced in prior work, we follow the definitions provided by Houssiau et al. [28] for the black- and white-box threat models as they are meant specifically for SDGs. We will assume, in *all* threat models, that the adversary can choose a worst-case target record and has knowledge of the neighboring datasets D and D', as is standard for auditing DP mechanisms [29, 49, 51].

Black-Box. In the black-box setting, the adversary has access to the synthetic dataset, S, as well as to the specifications of the SDG algorithm, but, crucially, not to the trained generative model $\mathcal{G}$ the synthetic dataset is sampled from.

Although this is the most practical (i.e., the weakest) threat model we consider, it is not the adversarial capability assumed by the DP guarantees provided by DP-SDGs. This is because the generative model fitting function, $\mathrm{GM}(\cdot)$, typically satisfies DP. The DP guarantees then transfer to the entire SDG algorithm as per the post-processing theorem. Thus, they should, in theory, hold even if the adversary has access to $\mathcal{G}$, and not just S. Nevertheless, many synthetic data libraries use simple black-box attacks to evaluate the privacy of differentially private synthetic data [28, 58]. Hence, we include this setting to compare the effectiveness of different black-box attacks in the DP setting and evaluate the impact of reduced adversarial power on empirical privacy estimates.

Passive White-Box. Here, we assume the adversary has access to the trained generative model $\mathcal{G}$ and its internal parameters, in addition to the synthetic dataset S. Examples of MIAs in this setting include [26, 27].

Active White-Box. Finally, we consider an adversary with not only access to the trained generative model but can also *actively* manipulate training. This setting was introduced by Nasr et. al. [49] in the context of auditing differentially private stochastic gradient descent (in discriminative models). In other words, the active white-box adversary can insert arbitrary ("canary") gradients into the training process of a machine learning model.

Note that while DP-SDGs like DPWGAN [69] and PATE-GAN [32] involve machine learning, they typically use different optimization algorithms rather than stochastic gradient descent (e.g., RMSProp). Therefore, we include this threat model to audit the DP guarantees of DP-SDGs that use machine learning models (e.g., GANs, VAEs, diffusion models).

Worst-Case Dataset. We also consider a setting where the adversary can choose worst-case neighboring datasets D and D'. Recall that DP guarantees hold not only for average-case neighboring datasets but also for worst-case ones [19]. While Nasr et al. [49] show that, for discriminative models, tight empirical estimates can be achieved even for average-case datasets, MIAs are generally harder for generative models [26], thus motivating us to consider both average and worst-case settings in our evaluation.

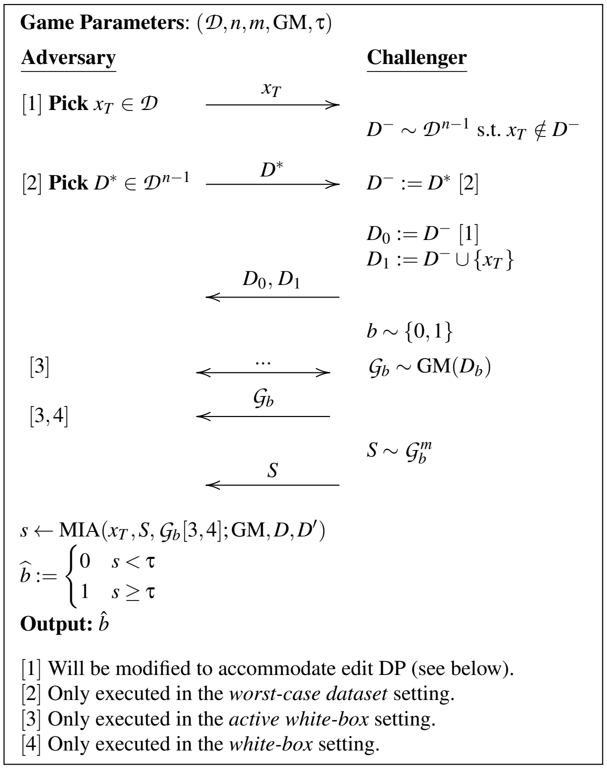

Game Parameters: $(\mathcal{D}, n, m, \mathrm{GM}, \tau)$
Adversary
Challenger
[1] **Pick** $x_T \in \mathcal{D}$
x_T
$D^- \sim \mathcal{D}^{n-1}$ s.t. $x_T \notin D^-$
[2] **Pick** $D^* \in \mathcal{D}^{n-1}$
D^*
$D^- := D^*$ [2]
$D_0 := D^-$ [1]
$D_1 := D^- \cup \{x_T\}$
D_0, D_1
$b \sim \{0,1\}$
[3]
$\cdots$
$\mathcal{G}_b \sim \mathrm{GM}(D_b)$
[3,4]
$\mathcal{G}_b$
$S \sim \mathcal{G}_b^m$
S
$s \leftarrow \mathrm{MIA}(x_T, S, \mathcal{G}_b[3,4]; \mathrm{GM}, D, D')$
$\hat{b} := \begin{cases} 0 & s < \tau \\ 1 & s \geq \tau \end{cases}$
Output: $\hat{b}$
[1] Will be modified to accommodate edit DP (see below).
[2] Only executed in the *worst-case dataset* setting.
[3] Only executed in the *active white-box* setting.
[4] Only executed in the *white-box* setting.

Figure 1: Distinguishability Game between Adversary and Challenger for add/remove DP, given a raw dataset ($\mathcal{D}$), the number of records in the original dataset (n), the number of records in the synthetic dataset (m), the generative model fitting function (GM), and a decision threshold τ.

3.3 DP Distinguishing Game

In Figure 1, we formalize the distinguishing game, played between an Adversary and a Challenger, used to audit the (add/remove) DP guarantees of a given DP-SDG algorithm.

While the game is for the add/remove DP definition, some SDGs – e.g., PrivBayes [75] – satisfy the edit DP definition instead; to this end, we modify the game to audit SDG algorithms that satisfy edit DP as follows. The Adversary chooses a worst-case pair of target records x_T and y instead of only x_T. Next, the Challenger sets $D_0 := D^- \cup \{y\}$ rather than $D_0 := D^-$. Doing so ensures that the Adversary distinguishes between $D^- \cup \{x_T\}$ and $D^- \cup \{y\}$.

3.4 Worst-Case Target Record

As mentioned, DP provides guarantees not only for the *average-case* but also *worst-case* target record. Thus, all adversaries have the ability to choose the target record. However, naïvely evaluating the privacy guarantees for the worst-case record would require every possible record in the domain to

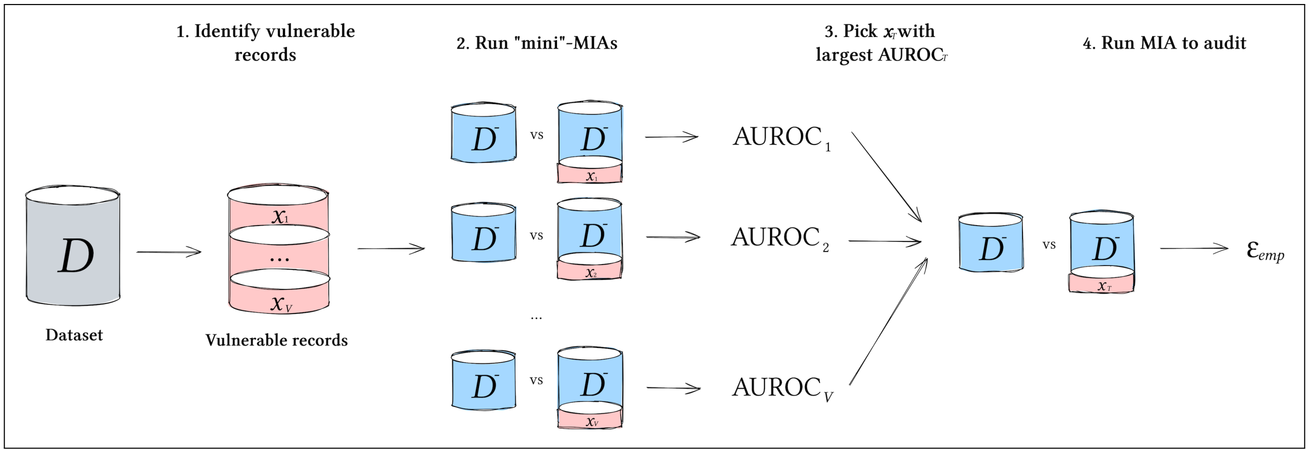


Figure 2: Choosing the worst-case target record to audit.

be audited; in practice, this is infeasible given the number of records in common high-dimensional datasets.

Rather, we let the Adversary use the vulnerable record identification procedure by Meeus et al. [46] to first select a subset of V most vulnerable target records from the original dataset. She then runs “mini” membership inference attacks on these V records and evaluates the area under the receiver operating characteristic curve (AUC) for each record. We call these “mini”-MIAs as they are only run over a small number of repetitions for each record (thus, they cannot be used to audit the DP guarantees with high confidence) and use them to let the Adversary choose the record with the highest AUC as the worst-case target record during the audit.

We find that $V = 100$ vulnerable records and 64 repetitions of each “mini”-MIA are enough to identify the worst-case target record. This process is outlined in Figure 2.

3.5 Worst-Case Neighboring Datasets

Similar to the choice of target record, DP provides guarantees not only against *average-case* (aka *natural*) neighboring datasets but also against the *worst-case* pair. While attacks are typically evaluated against the former [26, 28, 59], DP violations may not necessarily occur in these settings. Furthermore, leakage from target records may not be maximized, leading to loose empirical estimates that are far from the theoretical guarantees. Therefore, when auditing DP, it makes sense also to consider the worst-case neighboring datasets [38, 51].

However, worst-case neighboring datasets are likely algorithm-dependent, as leakage from different algorithms might be maximized in different settings. Designing worst-case datasets may also not be trivial in practice as there may be edge-case inputs the program may fail to execute. While this behavior technically constitutes a DP violation in itself, the audit, similar to error handling in compilers, should aim to identify as many DP violations as possible and not just stop at one. Thus, we experiment with different kinds of worst-case neighboring datasets to determine which properties of datasets maximize privacy leakage for different DP-SDGs.

We begin by using small neighboring datasets with very few records. While Nasr et al. [51] use $D' = \varnothing$, when auditing deployed SDG algorithms, none of the implementations studied cover this edge case and, in fact, they even fail to generate synthetic data. While this is a DP violation in itself, for the purposes of finding other violations that may also be present, we select neighboring datasets that are as small as possible without running into trivial runtime issues. Thus, we select $|D^-| = 2$, which we find to work for almost all SDG implementations.[1] We then experiment with two different properties of the worst-case datasets, namely, “narrow” datasets and repeating the target record.

Narrow Datasets. In theory, datasets with a large number of columns (i.e., wide datasets) increase the dimensionality of the generative model and may provide the Adversary with more signals (e.g., # of queries) that can be exploited by the MIA. However, DP mechanisms might make each signal more noisy, thus reducing the utility of the data and making it harder to exploit the signal [23]. As we do not know how the number of columns will affect the tightness of the empirical estimates, we test our attack against narrow datasets containing only 3 columns along with the original wide datasets.

Repeating the Target Record. MIAs are typically evaluated against a “fresh” record x_T that is only present in D (i.e., $x_T \in D$, $x_T \notin D'$). However, in some cases (especially in the context of anonymization), the *number of times* a target record appears in D (i.e., multiplicity) can reveal sensitive information about the dataset (e.g., homogeneity attack [40] against k-anonymity [63]). Thus, we consider the setting where x_T appears once in D and twice in D' to cover this edge case.

[1] For the DPWGAN (Synthcity) implementation, we set $|D^-| = 4$ as it fails to generate synthetic data for datasets with only two records.

4 Evaluation Framework

In this section, we discuss our experimental setup, introducing the datasets, synthetic data generation algorithms, and the membership inference attacks we use.

4.1 Datasets

We experiment with two tabular datasets used to train synthetic data generation (SDG) algorithms, which have been used extensively in prior work on synthetic data [8, 11, 37, 43] as well as in the 2018 NIST Synthetic Data Challenge [54]:

1. **Adult** [34], used to predict whether income exceeds \$50K from Census data. To make sure the dataset can be used as input to all DP-SDGs, we trim and bin the dataset to 11 categorical attributes (*age*, *workclass*, *education*, *marital-status*, *occupation*, *relationship*, *race*, *gender*, *hours-per-week*, *native-country*, and *income*).

2. **San Francisco Fire Dept Calls for Service (Fire)** [15], which records fire units' responses to calls made to them in 2016. It was used in the 2018 NIST Synthetic Data competition. Following prior work [6], we trim the dataset from 32 to 10 categorical attributes (*ALS Unit*, *Call Type Group*, *Priority*, *Call Type*, *Zipcode of Incident*, *Number of Alarms*, *Battalion*, *Call Final Disposition*, *City* and *Station Area*) to reduce the computational cost of generating thousands of synthetic datasets.

We focus on categorical datasets, primarily as many tabular DP-SDGs can only take those in input (e.g., PrivBayes [75], MST [43], AIM [44], RAP [8], and GEM [37]) and continuous values are often binned to categorical values (up to the practitioners' discretion) to be used with these DP-SDGs.

4.2 DP-SDG Algorithms

In this work, we experiment with the algorithms that participated in and won the 2018 NIST Differentially Private Synthetic Data Challenge competition [54]. We do so both due to their relevance and because these algorithms and their original implementations were independently verified by a team of experts to ensure the lack of privacy violations [43]. In other words, there is greater confidence that they do satisfy differential privacy. Yet, subtle DP violations, such as floating point bugs, have already been identified [38], which further motivates the need to audit these implementations.

In particular, we focus on three of the top five submissions to the NIST competition: PrivBayes [75], MST [43], and DPWGAN [4]. These have been popularly used in both research [56, 58] and industry [21], and encapsulate the different "paradigms" [6] of synthetic data generation.

For MST and DPWGAN, we audit the original implementations used in the NIST competition, which are publicly available on GitHub [52], and refer to them as **MST (NIST)** and **DPWGAN (NIST)**, respectively.[2] However, as PrivBayes is not included in the repository, we audit the popular publicly available implementation from the DataSynthesizer repository [56], which has been extensively used in prior work [6, 60], and refer to this as **PrivBayes (DS)**.

Method (Implementation)	DP Variant	Neighboring Dataset	Auditing Method
PrivBayes (DS) PrivBayes (Hazy)	ε-DP	Edit	(ε, δ)-DP
MST (NIST) MST (Smartnoise)	(ε, δ)-DP	Add/Remove	μ-GDP
DPWGAN (NIST) DPWGAN (Synthcity)	(ε, δ)-DP	Add/Remove	μ-GDP

Table 2: Algorithms (and implementations) audited.

Besides the three original implementations, we also audit three newer re-implementations. Given the algorithms' popularity, many companies and research labs have since included (and potentially modified) the algorithms in their software suites. However, as these modifications have not been independently verified, they may contain mistakes and privacy violations, once again prompting the need to audit them. More precisely,we audit **PrivBayes (Hazy)** [42], **MST (Smartnoise)** [47], and **DPWGAN (Synthcity)** [58].

A summary of all the algorithms tested and auditing methods used is reported in Table 2. Note that the PrivBayes implementations can only be audited using (ε, δ)-DP, with $\delta = 0$, as the underlying Laplace mechanism does not satisfy μ-GDP.

4.3 MIA Instantiations

4.3.1 Black-Box

For black-box audits, we focus on two attacks widely used as a measure of privacy leakage from the synthetic data.

Distance to Closest Record (DCR). DCR is a popular heuristic used by many software libraries [28, 55, 58] and companies [3, 61, 64, 66, 71]. Intuitively, synthetic data is expected to cause privacy leakage if it contains samples that are too close to the training dataset. Formally, given synthetic data S and target record x_T, the MIA outputs the score $-\min_{x \in S} d(x, x_T)$, for some distance metric d.

In our experiments, we first one-hot encode categorical features and use the Euclidean distance metric as done in prior work [28]. Furthermore, we make the score *negative* to ensure that a larger "score" corresponds to the presence of the target record in D and is consistent with our distinguishing game that outputs $\hat{b} = 1$ if and only if $s \geq \tau$.

Querybased. This attack [28] builds on shadow modeling techniques. Prior work using it include [25, 28, 46]. First, the

[2] In the rest of the paper, we use the Algorithm (Implementation) notation to denote the algorithm and its corresponding implementation we audit.

adversary generates many shadow synthetic datasets from D ($S_1, ..., S_n$) and D' ($S'_1, ..., S'_n$). Then, she evaluates the answers to queries targeted at x_T from the shadow synthetic datasets as features. We then train a Random Forest meta-classifier on these features to distinguish between synthetic datasets generated from D and D'. Finally, the adversary extracts the answers from the target synthetic dataset S and returns the output of the meta-classifier on its features as the score.

4.3.2 White-Box

Unlike black-box attacks that exploit privacy leakage from the synthetic datasets, white-box attacks exploit the leakage from the trained generative model parameters directly. As DP-SDG algorithms include a variety of generative models, ranging from simple statistical models to complex neural network architectures, the same attack cannot be generally applied to all algorithms. Therefore, we develop and instantiate different attacks for the different DP-SDG algorithms.

PrivBayes & MST. We develop a simple novel white-box attack against PrivBayes and MST that uses the shadow modeling technique. First, the adversary generates many shadow generative models $\mathcal{G}_1, ..., \mathcal{G}_n$ such that $\forall i\ \mathcal{G}_i \sim \text{GM}(D)$ and $\mathcal{G}'_1, ..., \mathcal{G}'_n$ such that $\forall i\ \mathcal{G}'_i \sim \text{GM}(D')$. Then, she extracts a set of *white-box* features from each of the shadow generative models. We experiment with two such features, namely, $\mathcal{F}_{naive}$ and $\mathcal{F}_{error}$. In the former, the adversary simply extracts the model parameters (joint conditional probability distributions for PrivBayes and marginals for MST) directly. In the latter, she first calculates the difference between each value in the model parameter and the corresponding exact value in D and sums these differences together. For PrivBayes and MST, each model parameter corresponds to a "measurement" (aka query) on the original dataset. Intuitively, this feature set represents the total error in the "noisy measurements" assuming D was the original dataset on which the generative model was fitted.

The adversary then trains a (Random Forest) meta-classifier on the extracted white-box features and assigns the output of the trained meta-classifier on the target generative model's extracted features as the score. In our experiments, we find that the $\mathcal{F}_{naive}$ feature set works best for PrivBayes, while, for MST, the $\mathcal{F}_{error}$ feature set produced marginally tighter guarantees (see Appendix A). Therefore, in the rest of this work, we use $\mathcal{F}_{naive}$ for PrivBayes and $\mathcal{F}_{error}$ for MST.

DPWGAN. For DPWGAN, we instantiate the LOGAN attack by Hayes et al. [26]. Intuitively, if the trained DPWGAN model overfits on a sensitive dataset, the discriminator will assign a higher confidence to records from that sensitive dataset. Thus, the adversary uses the output of the trained discriminator on the target record as the score.

4.3.3 Active White-Box

As mentioned, we only consider the active white-box attack against DPWGAN. In this setting, we instantiate the gradient canary attack by Nasr et al. [51]. Intuitively, since the adversary can manipulate the training process of the target generative model, at each iteration, she replaces the target record's actual gradient with a *canary gradient*. Next, the gradients of each record are clipped, aggregated, and noised to satisfy DP. The adversary calculates the dot product of the noised gradient update and the canary gradient to obtain an observation at each iteration. Finally, she sums these observations to derive a "score" for the target generative model.

While Nasr et al. [51] show that the active white-box attack produces tight guarantees when auditing discriminative models, we make a few modifications to the attack so that it can be applied to generative models. Unlike discriminative models, generative ones like GANs consist of multiple models (namely, a generator and a discriminator) trained in tandem. Since only the discriminator is trained with DP, in our work, we insert the canary gradient only in the discriminator, instead of the entire model architecture. More precisely, we insert the Dirac canary gradient (i.e., a gradient with zeros everywhere except a single index) as this produces the tightest empirical estimates in practice [49].

Also, models are often trained using software libraries like Opacus [73] or TensorFlow Privacy [24], which may not readily expose the aggregated gradient for users to audit. Therefore, we extract the gradients by calculating the difference in model parameters before and after a single iteration of training. Although this might lead to additional terms contributing to the gradient update (e.g., the RMSProp optimizer adds a moving average to the gradient update), from a software auditing point of view, we find this method more practical to implement, and it also remains effective in producing tight empirical guarantees. We illustrate the gradient canary attack [49, 51] with our adaptations to the DPWGAN model in Algorithm 1 highlighting the changes made by the adversary to the training algorithm in red (e.g., in lines 10 to 12, the adversary replaces the gradient of the target record with a canary gradient).

5 Experimental Results

This section presents our experimental evaluation geared to audit six state-of-the-art DP-SDG implementations with different, increasingly stronger threat models. First, we compare black-box attacks and analyze the differences in empirical guarantees (ε_{emp}) with *average-case* and *worst-case* neighboring datasets. We then experiment with white-box attacks (namely, LOGAN [26] for DPWGAN and a novel attack for PrivBayes and MST) as well as an *active* white-box one (adapting Nasr et al. [49]'s attack to the generative setting). Finally, we investigate whether our auditing procedure can identify common DP violations and discover new DP ones.

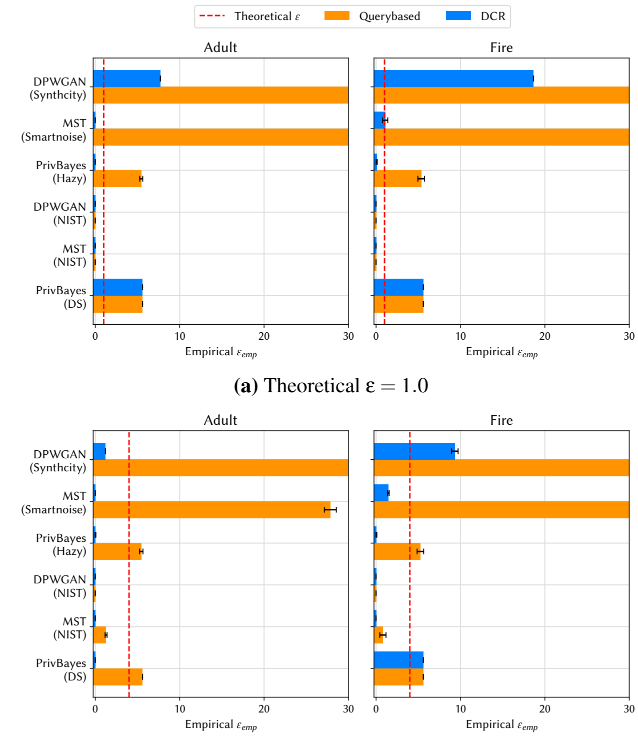
For each experiment, we train 10,000 SDG models; we use 6,000 as shadow models to train the meta-classifier for attacks that use shadow modeling (for attacks that do not, we do not

Algorithm 1 Active white-box auditing of DPWGAN

```
Require: Target record, $x_T$. Canary gradient, $g'$. Learning rate, $\alpha$. Clipping parameter, $c$. Batch size, $m$. Number of iterations of the critic per generator iteration, $n_{\text{critic}}$. Noise scale, $\sigma$. Group size, $L$. Gradient Norm bound, $c_p$.
Require: $w_0$, initial critic parameters. $\theta_0$, initial generator's parameters.
score ← 0
for $t \in [T]$ do
    for $i = 0, \ldots, n_{\text{critic}}$ do
        $w_{start} \leftarrow w$
        Pick a random sample $L_{t,i} = \{x^{(j)}\}_{j=1}^{L} \sim P_{data}(x)$
        from the real data
        Sample $\{z^{(j)}\}_{j=1}^{L} \sim p(z)$ a batch of prior samples
        ▷ Compute the per-example gradient
        $g_w(x^{(j)}) = \nabla_w f_w(x^{(j)})$ for $x^{(j)} \in L_{t,i}$
        $g_w(z^{(j)}) = \nabla_w f_w(G(z^{(j)};\theta))$ for $j \in [L]$
        if $x_T \in L_{t,i}$ then
            $g_w(x_T) = g'$
        end if
        ▷ Clip gradients
        for $x^{(j)} \in L_{t,i}$ do
            $\bar{g}_w(x^{(j)}) = g_w(x^{(j)}) / \max(1, \frac{||g_w(x^{(j)})||_2}{c_p})$
        end for
        for $j \in [L]$ do
            $\bar{g}_w(z^{(j)}) = g_w(z^{(j)}) / \max(1, \frac{||g_w(z^{(j)})||_2}{c_p})$
        end for
        ▷ Add Noise
        $\tilde{g}_w = \frac{1}{L}\left(\sum_{j=1}^{L} \bar{g}_w(x^{(j)}) + \mathcal{N}(0, \sigma^2 c_p^2 \mathbf{I})\right) -$
        $\frac{1}{L}\sum_{j=1}^{L} \bar{g}_w(z^{(j)})$
        $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, \tilde{g}_w)$
        $w \leftarrow \text{clip}(w, -c, c)$
        $w_{diff} \leftarrow w - w_{start}$
        score ← score $+ \langle w_{diff}, g' \rangle$
    end for
    Sample $\{z^{(j)}\}_{j=1}^{L} \sim p(z)$ a batch of prior samples
    $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{j=1}^{m} f_w(G(z^{(j)};\theta))$
    $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end for
return score, $\theta$, $w$
```

train any shadow models), and 2,000 models to choose the optimal threshold yielding the largest lower bound ε_{emp}. We then test all attacks on the remaining 2,000 models and calculate the false positive and false negative rates needed for the ε_{emp} estimation. Following prior work [49, 51], all lower bounds are given with 95% confidence (Clopper-Pearson [13]). We also report error bars, which we obtain via five-fold cross-validation – i.e., we split the 10,000 models into 5 partitions of 2,000 models each and repeatedly test the attack on each of the five partitions using the other four partitions to train the meta-classifier and choose the optimal threshold.

The source code needed to reproduce our experiments is available from https://github.com/spalabucr/synth-audit.



(a) Theoretical $\varepsilon = 1.0$

(b) Theoretical $\varepsilon = 4.0$

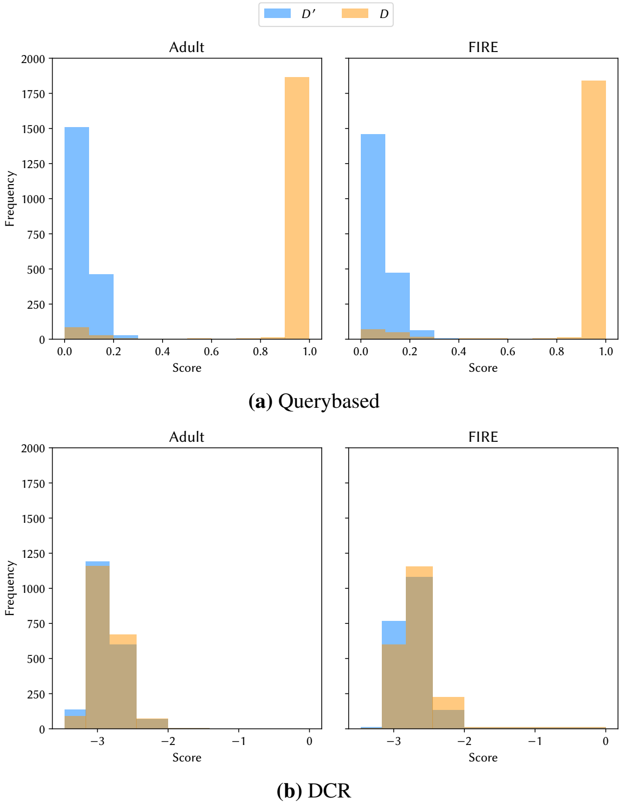
Figure 3: Black-box auditing, Querybased and DCR attacks.

5.1 Black-Box Auditing

5.1.1 Average-Case Dataset

We start by auditing the DP-SDG implementations by instantiating the DP distinguishing game (Section 3.3) with an average-case dataset ($|D| = 1000$) and two popular black-box attacks, i.e., Querybased and DCR [28]. In Figure 3, we report the empirical ε_{emp} guarantees for the six DP-SDG implementations, using Adult and Fire datasets, at two ε values corresponding to high and moderate privacy – respectively, $\varepsilon = 1.0$ and $\varepsilon = 4.0$.

DP Violations. We observe that the empirical privacy leakage is much larger than the theoretical guarantee (i.e., $\varepsilon_{emp} \gg \varepsilon$) for *all* the implementations not submitted to the NIST competition, i.e., DPWGAN (Synthcity), MST (Smartnoise), PrivBayes (Hazy), and PrivBayes (DS). By manually inspecting the code, we find that these directly extract the metadata from the input dataset. This "metadata violation" occurs as metadata like categories, minimum/maximum numerical value, etc., might unexpectedly leak information, especially for vulnerable target records with rare values [60]. Although this violation was already identified in 2022 for numerical datasets in PrivBayes (DS) and PATE-GAN [32], it still remains unfixed in the DataSynthesizer library. The same infringement also occurs in other implementations, such as PrivBayes (Hazy), MST (Smartnoise), and DPWGAN (Syn-



(a) Querybased

(b) DCR

Figure 4: Distribution of Querybased/DCR attack scores against PrivBayes (Hazy) trained on D vs D' at $\varepsilon = 1.0$.

thcity). Note that we reported these and all other violations to the respective library authors; see Section 7.6.

While Querybased identifies violations in 16 out of 24 experiments, DCR only identifies 7 of them. For the 9 violations identified by Querybased but not by DCR, the AUC of the latter is close to random (≈ 0.5), while that of the former is ≥ 0.95. In other words, DCR not only misses privacy violations but also severely underestimates privacy leakage from synthetic data. Evidently, it is ineffective at providing an effective measure of privacy and, in practice, should not be used to evaluate (differentially private) synthetic data.

Querybased vs DCR. We then investigate *why* this may be happening. To do so, we plot the raw scores output by the DCR and Querybased attacks against the PrivBayes (Hazy) implementation in Figure 4. Recall from Section 3.3 that these represent the confidence the attack assigns to SDG being trained on D; specifically, Querybased outputs a *probability* score (i.e., $s \in [0,1]$), whereas DCR outputs a *distance*, which we make negative (i.e., $s \in (-\infty, 0]$), as discussed in Section 4.3.1. Regardless, for both attacks, higher scores represent stronger confidence.

For Querybased, the distinct score separation when the DP-SDG is fitted on D and D' indicates that the meta-classifier learns and exploits the queries targeted at the target record effectively. For DCR, the distances between the target record and the closest synthetic record remain relatively similar regardless of whether the DP-SDG was fit on D or D'. As DCR relies on whole target records being memorized and output by the SDG, it seems unable to exploit more complex ways in which information can be leaked from synthetic data [6].

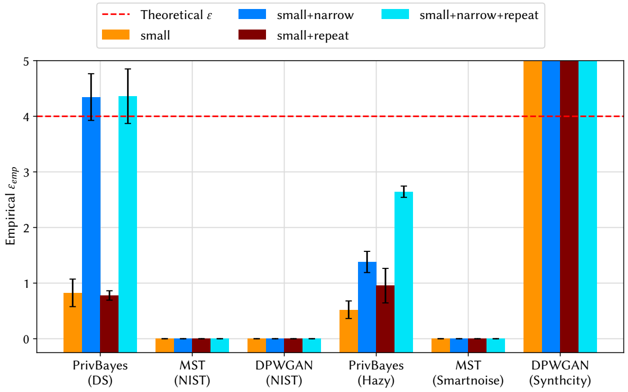


Figure 5: Black-box auditing at $\varepsilon = 4.0$ with different *worst-case* datasets using the Querybased attack.

Loose Estimates. Finally, auditing using black-box attacks results in several empirical privacy estimates much smaller than the theoretical upper bounds (i.e., $\varepsilon_{emp} \ll \varepsilon$). Interestingly, this happens for the DP-SDG implementations submitted to NIST, as $\varepsilon_{emp} \approx 0$ for MST (NIST) and DPWGAN (NIST), with both $\varepsilon = 1.0$ and 4.0. Arguably, it is unclear whether this is due to 1) leakage not being maximized under average-case neighboring datasets or 2) state-of-the-art black-box attacks being limited in power. To answer this, we next evaluate black-box attacks using *worst-case* neighboring datasets.

5.1.2 Worst-Case Datasets

To avoid the metadata violation discussed above, we craft worst-case neighboring datasets such that the domain of D is the same as D' (thus, the metadata extracted will be the same). Additionally, as the exact worst-case dataset could potentially be algorithm-dependent, we use worst-case datasets with a small number of rows (`small`) and experiment with two properties, i.e., 1) a small number of columns (`narrow`) and 2) repeating the target record (`repeat`). Finally, to focus on auditing the underlying noise-addition mechanism of the implementations, we use a provisional dataset to standardize the “structure” of the DP-SDGs, as done by the top NIST submissions [43]. More precisely, in the rest of the experiments, we standardize the Bayesian network built by PrivBayes and the marginals selected by MST across all models. In Figure 5, we plot the empirical guarantees (ε_{emp}) obtained by auditing using Querybased, for different worst-case datasets at theoretical $\varepsilon = 4.0$.

DP Violation. We find a DP violation for DPWGAN (Synthcity), regardless of the type of worst-case dataset. Recall that we prevent metadata violations by design (see Section 5.1);

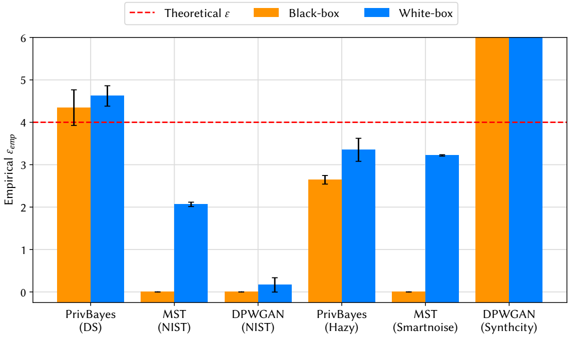


Figure 6: White-box vs black-box auditing at $\varepsilon = 4.0$ using implementation-specific worst-case neighboring datasets.

thus, this must be caused by something else. After manually inspecting the source code, we found that a random seed was re-used by the library for reproducibility, but this was not mentioned in the sample code. This removes any randomization from the code, thus making it deterministic, which results in a major DP violation. Alas, this type of bug was also found in other DP libraries, e.g., JAX [31].

Implementation-dependent worst-case. Next, we find that, even for the same algorithm, the worst-case dataset can be specific to the *implementation*. For PrivBayes (Hazy), the `small+narrow+repeat` dataset yields the highest privacy leakage estimate ($\varepsilon_{emp} = 2.64$), much higher than the estimate for `small+narrow` ($\varepsilon_{emp} = 1.38$). Whereas for PrivBayes (DS), the estimates are roughly the same for `small+narrow+repeat` and `small+narrow` datasets ($\varepsilon_{emp} = 4.36$ and 4.35, respectively). Ostensibly, this is due to each implementation introducing specific additional steps (pre-processing, validation, etc.), which might subtly alter the overall privacy leakage of the implementation itself.

Zero ε_{emp}. Finally, for MST (NIST), MST (Smartnoise), and DPWGAN (NIST), we see that $\varepsilon_{emp} \approx 0$ for all worst-case datasets. This suggests that even in the worst-case setting, state-of-the-art black-box MIAs are not powerful enough to exploit the privacy leakage from these DP-SDG implementations. DP-SDGs' theoretical guarantees typically apply to the underlying generative models directly and only transfer to the generated synthetic data through the post-processing theorem of DP; thus, loose estimates could also be due to the inability of state-of-the-art black-box MIAs to fully exploit the information available from DP-SDGs, which motivates us to explore stronger threat models.

5.2 White-Box Auditing

We now move on to adversaries with stronger capabilities, i.e., with access to the (final) fitted generative model. Specifically, we use the LOGAN [26] white-box attack against DPWGAN and a novel one against PrivBayes and MST.

First, we determine if white-box attacks can exploit the additional information available compared to black-box attacks.

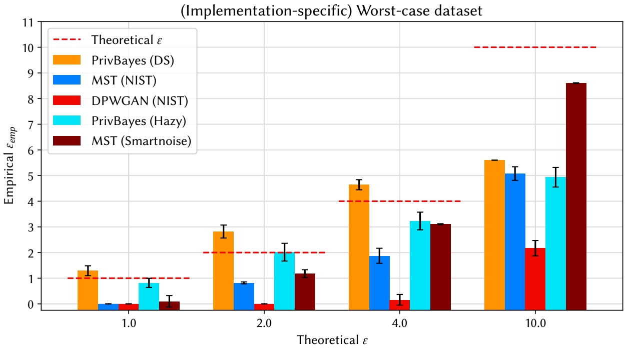


Figure 7: White-box auditing at $\varepsilon = 1.0, 2.0, 4.0, 10.0$.

Specifically, we compare the ε_{emp} values obtained with the white-box attacks vs. the Querybased black-box one in Figure 6. Experiments consider DP-SDGs trained with $\varepsilon = 4.0$ on the worst-case pair of neighboring datasets that produces the largest empirical guarantees for each DP-SDG implementation (which we find empirically, as discussed in Appendix B).

White- vs Black-box. For almost all implementations, the white-box attacks result in significantly tighter ε_{emp} estimates, i.e., closer to the theoretical ε-s, than the black-box attack. This is particularly evident for MST (NIST) and MST (Smartnoise), where the former obtain $\varepsilon_{emp} = 2.06$ and 3.22, respectively, while the latter is unable to detect any leakage (i.e., $\varepsilon_{emp} \approx 0$). Similarly, for PrivBayes (DS) and PrivBayes (Hazy), white-box audits produce tighter estimates of 4.62 and 3.35, respectively, compared to 4.35 and 2.64 for black-box attacks.

Note that, in the white-box setting, $\varepsilon_{emp} > \varepsilon$ for PrivBayes (DS) indicates another DP violation in the DataSynthesizer library. This violation was not obvious in the black-box setting as the standard deviation of ε_{emp} was larger, whereas, in the white-box setting, the theoretical ε is well outside the standard deviation of ε_{emp}. As mentioned previously, PrivBayes (DS) includes a number of pre-processing steps, other than inferring the metadata, that may not satisfy DP. Thus, we believe this is the most likely cause of the DP violation here.

Impact of privacy parameter. Next, as DP-SDGs can be instantiated with different levels of privacy (typically depending on the use case), we investigate whether our white-box audits can produce tight estimates at different ε values; see Figure 7. The empirical estimates are tightest for PrivBayes (DS) and PrivBayes (Hazy) across most ε-s.[3] Furthermore, ε_{emp} values for MST (NIST) and MST (Smartnoise) grow consistently with increasing ε, which indicates that the white-box attacks do leverage the increasing privacy leakage in practice. However, the estimates are not as tight as those of the PrivBayes implementations, especially at smaller ε-s. This may be due to the domain compression techniques used by MST; these are more aggressively applied at smaller ε-s, and might result in

[3] Note that we do not consider PrivBayes (Hazy) at $\varepsilon = 2.0$ a privacy violation as it lies within the standard deviation of $\varepsilon_{emp} = 2.01 \pm 0.35$.

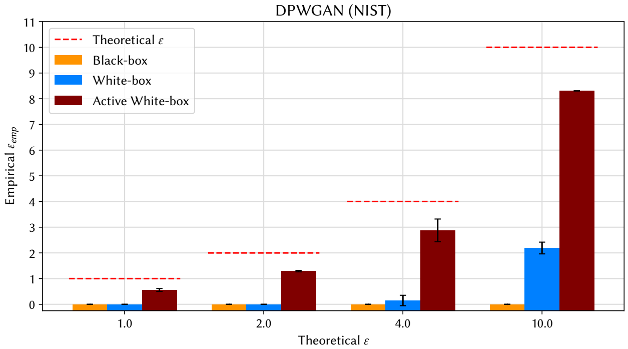


Figure 8: Black-box vs white-box vs active white-box auditing DPWGAN (NIST) at $\varepsilon = 1.0, 2.0, 4.0, 10.0$.

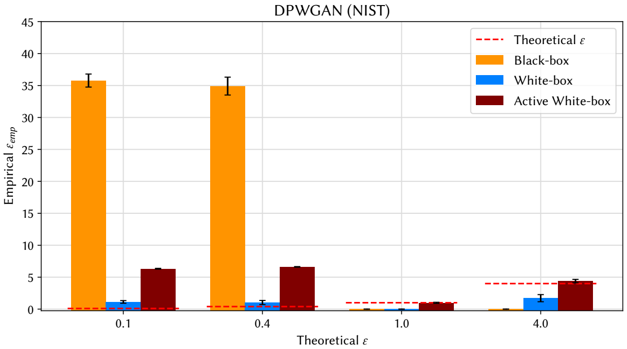


Figure 9: Black-box auditing of DPWGAN.

a loss of information, thus making it harder for white-box attacks to precisely estimate the privacy leakage. Nevertheless, using our white-box attacks, and worst-case dataset setting, we can audit the MST implementations much more tightly than prior work [28].

Finally, we find that white-box auditing does not produce tight ε_{emp} estimates for DPWGAN (NIST), even in the worst-case neighboring dataset setting. This indicates that not even the white-box adversary is powerful enough in this setting, thus motivating us to consider *active* white-box attacks.

5.3 Active White-Box Auditing

As discussed in Section 3.2, in the active white-box attack, the adversary manipulates training by inserting arbitrary gradients into the model, in this case, DPWGAN's discriminator.

In Figure 8, we report the resulting ε_{emp} estimates, using a worst-case neighboring dataset (`small+repeat`), to audit DPWGAN (NIST) at various theoretical ε-s . For completeness, we also report the ε_{emp} values for the black-box (Querybased) and white-box (LOGAN) attacks. We observe that the active attack produces relatively tight empirical ε_{emp} estimates, especially with large ε-s. Specifically, for $\varepsilon = 1.0, 2.0, 4.0, 10.0$, we obtain, respectively, $\varepsilon_{emp} = 0.56, 1.29, 2.88, 8.31$. This confirms that for DPWGAN, unlike PrivBayes and MST, auditing using the strongest active white-box attack is necessary to produce tight empirical estimates.

5.4 Finding Other DP Violations

The experimental analysis presented above allows us to identify the threat models and adversarial capabilities needed to tightly audit different DP-SDG implementations, highlighting the prevalence of metadata violations in DP-SDGs. Nonetheless, there could also be more subtle/less egregious violations. These are inherently harder to identify, and previous work had to rely on manual code inspection by experts to verify DP guarantees and find DP violations in DP-SDG implementations [54].

In the rest of this section, we investigate whether our auditing procedure can identify DP violations and verify DP guarantees *automatically*. In fact, using our auditing procedure, we identify a new violation in the implementation of DPWGAN submitted to NIST, along with violations that were previously found through manual inspection.

5.4.1 Early Stopping

Recall that the DPWGAN (NIST) implementation was submitted to the Differentially Private Synthetic Data Challenge competition [54]; participants had to submit their code and a technical report proving their algorithm satisfies DP [43]. The experts that reviewed both did not identify any violations, in other words, confirming that the implementation does satisfy (ε, δ)-DP. Our experiments presented so far also support this.

However, as seen earlier, tight empirical estimates may only be possible in *worst-case* settings. DPWGAN is a much more complex algorithm than PrivBayes/MST, involving many hyper-parameters that can be tuned. While we have only looked at worst-case target record and worst-case neighboring datasets thus far, for DPWGAN, we now look at worst-case *hyper-parameters* as well. After experimenting with different worst-case hyper-parameters, we find a DP violation when the `batch_size` hyperparameter is set to 1. In Figure 9, we plot the ε_{emp} estimates with `batch_size` set to 1, finding that, with small ε values (0.1, 0.4), $\varepsilon_{emp} \gg \varepsilon$. Specifically, auditing using the black-box attack with $\varepsilon = 0.1$ and 0.4 results in empirical estimates $\varepsilon_{emp} = 35.8$ and 34.9, respectively, while $\varepsilon_{emp} < \varepsilon$ for $\varepsilon = 1.0$ and 4.0.

We believe that the issue stems from the "early stopping" feature of the privacy accounting method. In the code, a privacy accountant tracks the privacy budget at each iteration, and training is aborted when that is exceeded. DPWGAN (NIST) applies two different accountants, depending on ε: for $\varepsilon < 0.7$, the privacy accountant is data-dependent as it uses the size of the dataset ($|D|$) without adding differentially private noise. Therefore, the model goes through a different number of iterations when trained on D and D', which is exploited by the black-box attack. Interestingly, the white-box and active white-box attacks do not detect a large DP violation in

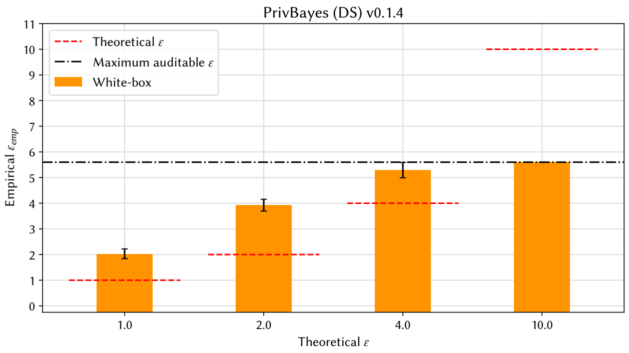


Figure 10: White-box auditing of DataSynthesizer v0.1.4.

this setting since they were only auditing the *discriminator*, whereas this bug affects the *generator* more significantly. As is not clear to us how to effectively attack the generator in the white-box/active white-box settings, we believe this prompts an interesting area for further research.

5.4.2 Noise Scale

One issue with DP-SDG implementations previously found through manual inspections is the noise scale bug, where the noise parameter of the algorithm is incorrectly configured. For instance, this was identified in the PrivBayes (DS) implementation at v0.1.4 and raised as a GitHub issue by a privacy researcher.[4] Unlike the early stopping bug, this one does not completely break the DP guarantees of the algorithm; rather, it results in the DP guarantee (slightly) overestimating the actual privacy protections. Thus, this type of bug can only be caught if the auditing procedure is tight (i.e., $\varepsilon_{emp} \approx \varepsilon$); otherwise, even if the actual ε parameter is much larger than what is claimed, the auditing procedure may not return $\varepsilon_{emp} \gg \varepsilon$, thus not flagging it as a DP violation.

Next, we investigate whether our auditing procedure can automatically detect the noise scale bug without the need for manual expert analysis. Specifically, we use our white-box attack to audit PrivBayes (DS) v0.1.4. Figure 10 shows the results of our white-box auditing procedure at various levels of theoretical ε. For $\varepsilon = 1.0$ and 2.0, our auditing procedure produces empirical estimates $\varepsilon_{emp} = 2.03$ and 3.92, respectively—clearly flagging the DP violation. It actually calculates the *magnitude* of the violation accurately, as the *true* ε is approximately twice the *claimed* ε.

However, at $\varepsilon = 4.0$, our procedure only identifies that the noise scale is configured wrongly but does not accurately calculate the magnitude of the error; at $\varepsilon = 10.0$, it does not even detect the violation as it reaches the *maximum auditable* ε limit. Recall from Section 2.2 that the maximum auditable ε limit is an inherent limitation of our auditing procedure.

5.5 Takeaways

Our experimental analysis shows that the state-of-the-art *black-box* MIAs commonly used to evaluate privacy in DP-SDGs may be severely limited in power. For instance, the DCR heuristic vastly underestimates privacy leakage from the synthetic data, often achieving AUCs close to a random guess (≈ 0.5), in settings where Querybased yields ≥ 0.95 AUC. With the latter, we also find metadata violations in many DP-SDG implementations (stemming from learning metadata directly from the input dataset). Nonetheless, auditing using Querybased still generally results in loose empirical estimates of privacy leakage, even in worst-case settings.

Arguably, to estimate the privacy leakage from DP-SDGs tightly, we need implementation-specific worst-case datasets and stronger threat models. Our experiments show that the privacy leakage of different DP-SDG implementations is maximized for different types of worst-case datasets. Auditing using a novel *white-box* attack yields tight estimates for PrivBayes and MST implementations; however, for machine learning models like DPWGAN, we need attacks in a much more powerful, *active white-box* threat model. When auditing PrivBayes (Hazy), MST (Smartnoise), and DPWGAN (NIST) at $\varepsilon = 4.0$ using our auditing procedure, we obtain nearly tight empirical privacy estimates of $\varepsilon_{emp} = 3.23$, 3.10, and 3.02, respectively. In comparison, using Querybased only achieves loose estimates of $\varepsilon_{emp} = 2.64$, 0.00, and 0.00, respectively.

Last but not least, our auditing procedure can find several DP violations (e.g., noise scale bug) in DP-SDGs *automatically*, without the need for manual inspection. It also identifies a new DP violation in the DPWGAN (NIST) implementation.

6 Related Work

DP-SDG Auditing. To our knowledge, our work presents the first large-scale audit of DP-SDG algorithms and implementations. Houssiau et al. [28] introduce, and audit DP-SDGs with, the TAPAS toolbox, though for only a single implementation of a single DP-SDG. They empirically estimate the privacy guarantees of MST [43] at $\varepsilon = 10$ using the black-box Querybased attack, but only achieve loose guarantees. By contrast, our audits include multiple algorithms/implementations, stronger threat models, and are considerably tighter.

MIAs against synthetic data. The DCR heuristic is one of the earliest black-box methods used for MIAs [27, 39], although having limited effectiveness. Stadler et al. [60] use shadow modeling to show that outlier records in synthetic data are often vulnerable to black-box MIAs. They also find DP metadata violations in implementations of PrivBayes and PATE-GAN (i.e., they extract metadata from the input dataset). However, their goal meaningfully differs from ours as they do not focus on auditing and only consider black-box attacks.

The only white-box MIA against SDGs is LOGAN [26], which attacks GANs (we use it for DPWGAN); we present the

[4] https://github.com/DataResponsibly/DataSynthesizer/issues/34

first white-box MIAs against PrivBayes [75] and MST [43].

Empirically Estimating Privacy in DP-ML. Prior work has extensively focused on empirically estimating the privacy of differentially private discriminative models (DP-ML) in both centralized and federated settings [5, 29, 30, 35, 41, 49, 51, 62, 68, 74]. Jayaraman et al. [30] and Jagielski et al. [29] present auditing schemes for DP-ML but generally only achieve loose empirical estimates. Nasr et al. [51] audits DP-ML using the (ε, δ)-DP definition and Clopper-Pearson intervals, requiring a million runs at $\varepsilon = 10$. Zanella-Béguelin et al. [74] focus on reducing the number of training runs using so-called *credible* intervals, while Nasr et al. [49] audit DP-ML using the μ-GDP definition and credible intervals and show that 1,000 runs are in fact enough to audit models at $\varepsilon = 10$. While auditing with 1,000 runs is generally feasible for centralized learning, it might be less so in resource-constrained settings typical of federated learning; thus, another line of work focuses on reducing the number of runs to one [5, 41, 62].

Tightly Auditing DP-ML. Nasr et al. [49] present a tight auditing scheme for discriminative models trained using differentially private stochastic gradient descent [1]. They show that *natural* (i.e., not adversarially crafted) datasets are enough for tightness, considering a white-box adversary who can choose arbitrary *canary* gradients at each step. Arguably, our work is broader in nature and scope. We consider three different training algorithms for generative models, compared to just stochastic gradient descent. Also, in discriminative models, there is a single signal (i.e., the model's loss on the target record) that can be exploited by MIAs, whereas generative models' outputs lie in a higher dimensional space, producing many possible signals. Thus, we experiment with multiple MIAs for each threat model (e.g., Querybased and DCR for black-box), studying the disparity of their effectiveness. Incidentally, we find that in DP-SDGs, unlike discriminative models, adversarially crafted implementation-specific worst-case datasets are necessary to achieve tightness.

Auditing DP Implementations. Prior audits of DP implementations include DP-Sniper [9], DP-Opt [53], and Delta-Siege [38]. Note that [9, 53] do not consider DP-SDGs, while [38] is orthogonal to our work as it aims to amplify existing distinguishers and classifiers to identify floating-point DP violations in DP implementations (including MST [43]).

7 Discussion & Conclusion

7.1 Summary

This paper focused on tightly auditing (six) differentially private synthetic data generation (DP-SDG) implementations. We analyzed the key factors affecting tightness, running several MIAs in different threat models and experimenting with worst-case datasets. Our analysis shows that the privacy leakage of DP-SDGs can indeed be tightly estimated empirically, but only for strong adversaries and worst-case neighboring datasets. In the process, we proposed novel white-box MIAs against PrivBayes and MST and presented an adaptation of Nasr et al. [49]'s gradient canary attack to DPWGAN.

Furthermore, our automated auditing procedure discovered DP violations in most DP-SDG implementations, including a new DP one in the DPWGAN implementation submitted to the NIST DP Synthetic Data Challenge. Overall, we are confident that our work will encourage more research into automated auditing tools so that DP-SDG implementations can be verified easily and at scale.

7.2 The Importance of Automated Auditing

Designing automated auditing tools is an important area of research as these enable researchers and practitioners to find bugs and violations of formal guarantees in real-world implementations. Arguably, this is particularly relevant in the context of Differential Privacy (DP), as DP is increasingly used in the wild to protect the privacy of real-world users [7, 17, 20, 45], as well as citizens in critical settings like the U.S. Census [2]. This extends to differentially private synthetic data generation (DP-SDG) tools, which are being deployed to protect the data of sensitive populations like the individuals in Microsoft's human trafficking dataset [48] or in healthcare settings [67]. Bugs in these production systems break these protections and enable adversaries to learn sensitive information about end users [22, 65].

This makes it crucial to audit algorithms and implementations as a systematic way to verify and guarantee the privacy of vulnerable groups in the wild. To this end, our work showcases how manual "inspection" by experts to find DP violations might miss some subtle violations; overall, manual analysis may not be scalable, as each version of a released DP-compliant software will have to be verified individually. Conversely, automated auditing can cover a wider range of violations and be included in continuous integration pipelines, thus reducing the potential for DP violations to be missed. Indeed, our experiments show that our auditing procedure can *automatically* find DP violations in DP-SDGs, including new ones that were previously missed.

7.3 Powerful Threat Models

As discussed, our auditing procedure goes beyond black-box threat models typically used in state-of-the-art MIAs against tabular synthetic data [60], considering more powerful ones – i.e., white-box, active white-box, and worst-case dataset attacks. Naturally, the stronger the threat models, the stronger the assumptions in place. In particular, white-box attacks are generally less practical to mount, as it is not always clear how the adversary can gain access to the final fitted generative model. Arguably, the active white-box and worst-case dataset attacks may be even less practical – e.g., the former assumes that the adversary can actively, yet possibly stealthily, manipulate model training. On the other hand, as argued in [26], white-box attacks can be considered practical when models are released following a data breach or when they are compressed/deployed to smartphones.

Nevertheless, we emphasize that the purpose of auditing is to ensure that the *provably correct* privacy guarantees of DP are not "lost" in practice – e.g., due to implementation bugs – regardless of the threat model. Furthermore, DP violations can sometimes result in realistic privacy leaks as well. For instance, our work shows that the metadata violation leads to a membership inference attack, in the black-box setting, with an AUC of ≥ 0.95 for PrivBayes (DS), PrivBayes (Hazy), MST (Smartnoise), and DPWGAN (Synthcity) (see Figure 3). Similarly, prior work [7] has also demonstrated that sensitive information (e.g., skin tone or political orientation) can leak from DP algorithms when empirical privacy guarantees do not match the intended theoretical ones. Finally, DP is, by design, a robust mathematical framework that provides privacy protections even against *worst-case* threat models, including the ones considered in this paper.

7.4 Computational Cost of Auditing

A potential concern with automated auditing is the computational cost incurred. Not only are state-of-the-art auditing tools affected by the number of models that have to be built, but they also depend on the computational efficiency of the individual implementations. For instance, in our experiments, it took 6.72s to generate a synthetic dataset from the (downsized) ADULT dataset with PrivBayes (Hazy) and more than 5x longer (39.0s) with PrivBayes (DS). Fitting the 10,000 models required for auditing took, respectively, 35 mins and 3 hours 24 mins for PrivBayes (Hazy) and PrivBayes (DS) by parallelizing the computation on a server with an Intel Xeon CPU with 32 2.20GHz cores and 128GB of RAM. DPWGAN and MST took longer, with their NIST implementations taking 4 hours 8 mins and 14 hours 18 mins, respectively.

While we believe this is ultimately reasonable, reducing the number of models needed for auditing could be interesting for future work. Incidentally, note that recent work [5, 62] has presented one-shot auditing techniques (i.e., only using one model); however, these methods are specific to auditing differentially private stochastic gradient descent and do not provide tight empirical guarantees.

7.5 Limitations & Future Work

Although our work succeeds in providing (almost) tight empirical estimates of privacy for the DP-SDG implementations studied, it is, naturally, not without limitations.

First, our auditing procedure requires thousands of synthetic datasets and models to be trained; this is due both to the use of shadow models, which trains a classifier on potentially thousands of samples, and the Clopper-Pearson confidence intervals limiting the maximum auditable ε. For a given number of test observations, even when an adversary can perfectly distinguish between $\mathcal{M}(D)$ and $\mathcal{M}(D')$, i.e., $\alpha = \beta = 0$, the 95% upper bounds $\overline{\alpha}$ and $\overline{\beta}$ are lower bounded. As these bounds are used to calculate ε_{emp}, this results in an upper bound on the ε_{emp} as well.

Second, we need millions of observations to audit at relatively large values of ε, such as $\varepsilon = 10$, using (ε, δ)-DP [51]. While using credible intervals from [74] could improve on this, we find that, for 2,000 test observations, the difference in maximum auditable ε is only 0.51. Auditing with μ-GDP requires much fewer observations (≈ 22) but can only be applied to mechanisms that satisfy μ-GDP, thus excluding pure DP mechanisms like PrivBayes.

Recent work [5, 62] propose techniques to audit DP discriminative models using only one trained model ("one-shot"). However, they do not provide tight empirical guarantees, and it is not clear how they can be applied to generative models. Therefore, we leave exploring these directions to future work.

In the future, we also plan to explore one or few-shot empirical privacy estimation of DP-SDGs and explore the deployment of our procedure into continuous integration pipelines.

7.6 Ethics & Disclosure

Our work does not involve attacking live systems or private datasets. In the spirit of responsible disclosure, in February 2024, we reported the five DP violations discussed in this paper to the respective library authors. We offered to clarify, assist in fixes, and provide initial suggestions and recommendations. We also refrained from making our findings public for at least 90 days from disclosure.

As of May 2024, only the authors of PrivBayes (Hazy) and DPWGAN (Synthcity) have responded to our disclosure. The PrivBayes (Hazy) library now displays a privacy warning to users when it automatically learns the metadata from the dataset. Unfortunately, the latest version of DPWGAN (Synthcity) (v0.2.10) still contains both the metadata and PRNG reuse violations. PrivBayes (DS), MST (Smartnoise), and DPWGAN (NIST) have not made any commits to their GitHub repository since then; thus, the violations are still present in the publicly available libraries.

Acknowledgments. This work has been supported by the National Science Scholarship (PhD) from the Agency for Science Technology and Research, Singapore.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *CCS*, 2016.

[2] John M Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson Garfinkel, Micah Heineck, Christine Heiss, Robert Johns, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, et al. The 2020 census disclosure avoidance system topdown algorithm. *Harvard Data Science Review*, 2022.

[3] Mostly AI. Read the QA Report. https://mostly.ai/docs/guides/qa-report#nearest-neighbor-distance-ratio, 2023.

[4] Moustafa Alzantot and Mani Srivastava. Differentially Private Dataset Release using Wasserstein GANs. https://github.com/nesl/nist_differential_privacy_synthetic_data_challenge/, 2019.

[5] Galen Andrew, Peter Kairouz, Sewoong Oh, Alina Oprea, H Brendan McMahan, and Vinith Suriyakumar. One-shot Empirical Privacy Estimation for Federated Learning. In *ICLR*, 2024.

[6] Meenatchi Sundaram Muthu Selva Annamalai, Andrea Gadotti, and Luc Rocher. A Linear Reconstruction Approach for Attribute Inference Attacks against Synthetic Data. *arXiv:2301.10053*, 2023.

[7] Apple. Learning with Privacy at Scale. https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf, 2017.

[8] Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. Differentially Private Query Release Through Adaptive Projection. In *ICML*, 2021.

[9] Benjamin Bichsel, Samuel Steffen, Ilija Bogunovic, and Martin Vechev. DP-Sniper: Black-Box Discovery of Differential Privacy Violations using Classifiers. In *IEEE S&P*, 2021.

[10] US Census Bureau. The Census Bureau's Simulated Reconstruction-Abetted Re-identification Attack on the 2010 Census. https://www.census.gov/data/academy/webinars/2021/disclosure-avoidance-series/simulated-reconstruction-abetted-re-identification-attack-on-the-2010-census.html, 2021.

[11] Kuntai Cai, Xiaoyu Lei, Jianxin Wei, and Xiaokui Xiao. Data Synthesis via Differentially Private Markov Random Fields. *VLDB Endowment*, 2021.

[12] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-Only Membership Inference Attacks. In *ICML*, 2021.

[13] Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 1934.

[14] European Commission. European data strategy. https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/european-data-strategy, 2022.

[15] DataSF. Fire Department Calls for Service. https://data.sfgov.org/Public-Safety/Fire-Department-Calls-for-Service/nuek-vuh3, 2016.

[16] Edoardo Debenedetti, Giorgio Severi, Nicholas Carlini, Christopher A Choquette-Choo, Matthew Jagielski, Milad Nasr, Eric Wallace, and Florian Tramèr. Privacy Side Channels in Machine Learning Systems. *arXiv:2309.05610*, 2023.

[17] Damien Desfontaines. A list of real-world uses of differential privacy. https://desfontain.es/privacy/real-world-differential-privacy.html, 2023.

[18] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian Differential Privacy. *arXiv:1905.02383*, 2019.

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, 2006.

[20] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *CCS*, 2014.

[21] Office for National Statistics. Synthesising the linked 2011 Census and deaths dataset while preserving its confidentiality. https://datasciencecampus.ons.gov.uk/synthesising-the-linked-2011-census-and-deaths-dataset-while-preserving-its-confidentiality/, 2023.

[22] Andrea Gadotti, Florimond Houssiau, Meenatchi Sundaram Muthu Selva Annamalai, and Yves-Alexandre de Montjoye. Pool Inference Attacks on Local Differential Privacy: Quantifying the Privacy Guarantees of Apple's Count Mean Sketch in Practice. In *USENIX Security*, 2022.

[23] Georgi Ganev, Kai Xu, and Emiliano De Cristofaro. Understanding how Differentially Private Generative Models Spend their Privacy Budget. *arXiv:2305.10994*, 2023.

[24] Google. TensorFlow Privacy. https://github.com/tensorflow/privacy, 2019.

[25] Florent Guépin, Matthieu Meeus, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. Synthetic is all you need: removing the auxiliary data assumption for membership inference attacks against synthetic data. *arXiv:2307.01701*, 2023.

[26] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: Membership Inference Attacks Against Generative Models. In *PETS*, 2019.

[27] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. In *PETS*, 2019.

[28] Florimond Houssiau, James Jordon, Samuel N Cohen, Andrew Elliott, James Geddes, Callum Mole, Camila Rangel-Smith, and Lukasz Szpruch. TAPAS: a Toolbox for Adversarial Privacy Auditing of Synthetic Data. In *SyntheticData4ML Workshoph NeurIPS*, 2022.

[29] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing Differentially Private Machine Learning: How Private is Private SGD? *NeurIPS*, 2020.

[30] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *USENIX Security*, 2019.

[31] Matthew Johnson. GitHub Pull Request: fix prng key reuse in differential privacy example. https://github.com/google/jax/pull/3646, 2020.

[32] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *ICLR*, 2018.

[33] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The Composition Theorem for Differential Privacy. In *ICML*, 2015.

[34] Ronny Kohavi and Barry Becker. UCI ADULT Data Set. https://archive.ics.uci.edu/ml/datasets/adult, 1996.

[35] Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso. Disparate vulnerability to membership inference attacks. In *PETS*, 2022.

[36] Ninghui Li, Zhikun Zhang, and Tianhao Wang. DPSyn: Experiences in the NIST Differential Privacy Data Synthesis Challenges. *arXiv:2106.12949*, 2021.

[37] Terrance Liu, Giuseppe Vietri, and Steven Z Wu. Iterative Methods for Private Synthetic Data: Unifying Framework and New Methods. In *NeurIPS*, 2021.

[38] Johan Lokna, Anouk Paradis, Dimitar I Dimitrov, and Martin Vechev. Group and Attack: Auditing Differential Privacy. In *CCS*, 2023.

[39] Pei-Hsuan Lu, Pang-Chieh Wang, and Chia-Mu Yu. Empirical Evaluation on Synthetic Data Generation with Generative Adversarial Network. In *WIMS*, 2019.

[40] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM TKDD*, 2007.

[41] Samuel Maddock, Alexandre Sablayrolles, and Pierre Stock. CANIFE: Crafting Canaries for Empirical Privacy Measurement in Federated Learning. In *ICLR*, 2023.

[42] Sofiane Mahiou, Kai Xu, and Georgi Ganev. dpart: Differentially Private Autoregressive Tabular, a General Framework for Synthetic Data Generation. *TPDP*, 2022.

[43] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST Contest: A scalable and general approach to differentially private synthetic data. *JPC*, 2021.

[44] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data. *VLDB Endowment*, 2022.

[45] Brendan McMahan and Abhradeep Thakurta. Federated learning with formal differential privacy guarantees. *Google AI Blog*, 2022.

[46] Matthieu Meeus, Florent Guepin, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. Achilles' Heels: Vulnerable Record Identification in Synthetic Data Publishing. *arXiv:2306.10308*, 2023.

[47] Microsoft. SmartNoise SDK: Tools for Differential Privacy on Tabular Data. https://github.com/opendp/smartnoise-sdk, 2021.

[48] Microsoft. IOM and Microsoft release first-ever differentially private synthetic dataset to counter human trafficking. https://www.microsoft.com/en-us/research/blog/iom-and-microsoft-release-first-ever-differentially-private-synthetic-dataset-to-counter-human-trafficking/, 2022.

[49] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight Auditing of Differentially Private Machine Learning. In *USENIX Security*, 2023.

[50] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning. In *IEEE S&P*, 2018.

[51] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning. In *IEEE S&P*, 2021.

[52] National Institute of Standards and Technology. Differential Privacy Synthetic Data Challenge Algorithms. https://github.com/usnistgov/PrivacyEngCollabSpace/tree/master/tools/de-identification/Differential-Privacy-Synthetic-Data-Challenge-Algorithms, 2024.

[53] Ben Niu, Zejun Zhou, Yahong Chen, Jin Cao, and Fenghua Li. DP-Opt: Identify High Differential Privacy Violation by Optimization. In *WASA*, 2022.

[54] National Institute of Standards and Technology. 2018 Differential Privacy Synthetic Data Challenge. https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/past-prize-challenges/2018-differential-privacy-synthetic, 2018.

[55] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The Synthetic data vault. In *IEEE DSAA*, 2016.

[56] Haoyue Ping, Julia Stoyanovich, and Bill Howe. DataSynthesizer: Privacy-Preserving Synthetic Datasets. In *SSDBM*, 2017.

[57] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. In *NDSS*, 2018.

[58] Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities. *arXiv:2301.07573*, 2023.

[59] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks against Machine Learning Models. In *IEEE S&P*, 2017.

[60] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic Data – Anonymisation Groundhog Day. In *USENIX Security*, 2022.

[61] Statice. Statice by Anonos. https://www.statice.ai/, 2023.

[62] Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy Auditing with One (1) Training Run. In *NeurIPS*, 2023.

[63] Latanya Sweeney. k-anonymity: A model for protecting privacy. *IJUFKS*, 2002.

[64] Syntegra. Syntegra. https://www.syntegra.io/, 2023.

[65] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12. *arXiv:1709.02753*, 2017.

[66] Tonic.ai. Tonic. https://www.tonic.ai/, 2023.

[67] UC Davis. $1.2 million to study synthetic data use. https://health.ucdavis.edu/health-magazine/issues/fall2022/noteworthy/study-synthetic-data-use.html, 2022.

[68] Chengkun Wei, Minghu Zhao, Zhikun Zhang, Min Chen, Wenlong Meng, Bo Liu, Yuan Fan, and Wenzhi Chen. DPMLBench: Holistic Evaluation of Differentially Private Machine Learning. In *ACM CCS*, 2023.

[69] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv:1802.06739*, 2018.

[70] Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. Assessing privacy and quality of synthetic health data. In *ACM AIDR*, 2019.

[71] YData. YData. https://www.ydata.ai/, 2023.

[72] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *CCS*, 2022.

[73] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen,

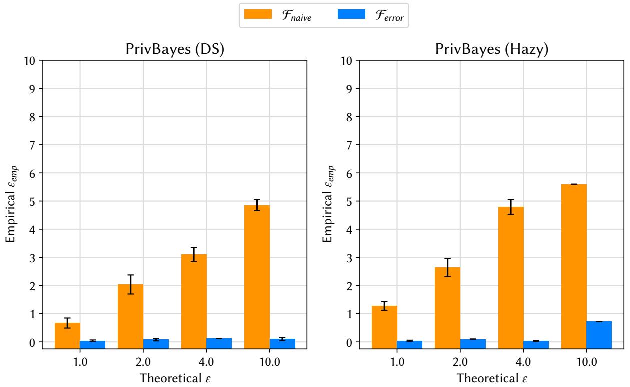


Figure 11: White-box auditing of PrivBayes implementations for different feature sets.

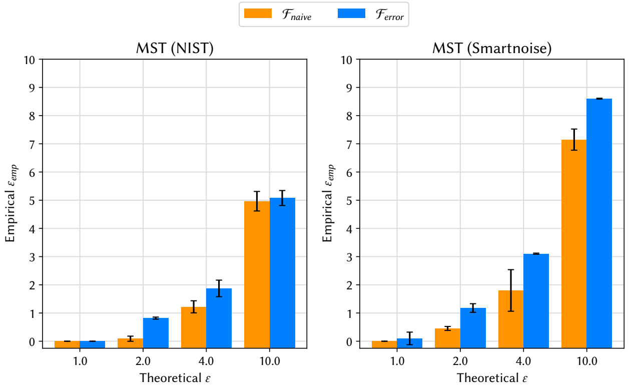


Figure 12: White-box auditing MST implementations for different feature sets.

Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in PyTorch. *arXiv:2109.12298*, 2021.

[74] Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Ahmed Salem, Victor Rühle, Andrew Paverd, Mohammad Naseri, Boris Köpf, and Daniel Jones. Bayesian estimation of differential privacy. In *ICML*, 2023.

[75] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. PrivBayes: Private Data Release via Bayesian Networks. *ACM TODS*, 2017.

A Comparing Feature Sets for White-Box Attacks

PrivBayes. In Figure 11, we plot the empirical ε_{emp} guarantees obtained when auditing PrivBayes using the implementation-specific worst-case dataset (see Appendix B) but for different white-box features ($\mathcal{F}_{naive}$ and $\mathcal{F}_{error}$) that are extracted from the fitted generative models. We find that the raw model parameters ($\mathcal{F}_{naive}$) result in much better guarantees than the error value feature set ($\mathcal{F}_{error}$) for all ε-s.

MST. In Figure 12, we plot the empirical ε_{emp} guarantees obtained when auditing MST using the implementation-specific worst-case dataset (see Appendix B) but for different white-box features ($\mathcal{F}_{naive}$ and $\mathcal{F}_{error}$) that are extracted from the fitted generative models. We find that the error value feature set ($\mathcal{F}_{error}$) results in marginally tighter guarantees for all ε-s.

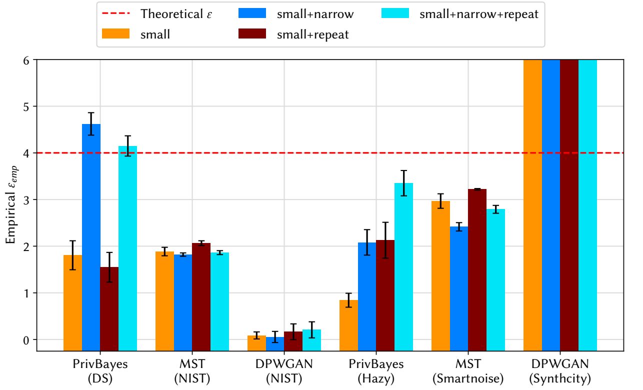


Figure 13: White-box auditing DP-SDG implementations at $\varepsilon = 4.0$ for different worst-case datasets.

B Comparing Worst-Case Datasets for White-box Auditing

Figure 13 compares the empirical ε_{emp} guarantees obtained by the white-box attacks (specific to the DP-SDG implementation) for different worst-case datasets. Similar to the black-box setting, we find that the worst-case dataset is implementation-dependent even for white-box attacks.

Specifically, for the PrivBayes (DS) and PrivBayes (Hazy) implementations, the `small+narrow` dataset and `small+narrow+repeat` dataset produce the tightest guarantees similar to the black-box setting. On the other hand, for the MST (NIST), MST (Smartnoise), and DPWGAN (NIST), the `small+repeat` dataset produces the tightest guarantees.