

USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

Sequential Auditing for f-Differential Privacy

Tim Kutta, *Aarhus University;* Martin Dunsche, *Ruhr University Bochum;*
Yu Wei and Vassilis Zikas, *Georgia Institute of Technology*

https://www.usenix.org/conference/usenixsecurity26/presentation/kutta

This paper is included in the Proceedings of the
35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the
35th USENIX Security Symposium
is sponsored by



جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

ARTIFACT EVALUATED
USENIX
AVAILABLE

Sequential Auditing for f-Differential Privacy

Tim Kutta[1]*, Martin Dunsche[2], Yu Wei[3], Vassilis Zikas[3]

[1]*Aarhus University*
[2]*Ruhr University Bochum*
[3]*Georgia Institute of Technology*

Abstract

We present new auditors to assess Differential Privacy (DP) of an algorithm based on output samples. Such empirical auditors are common to check for algorithmic correctness and implementation bugs. Most existing auditors are batch-based or targeted toward the traditional notion of (ε, δ)-DP; typically both. In this work, we shift the focus to the highly expressive privacy concept of f-DP, in which the entire privacy behavior is captured by a single tradeoff curve. Our auditors detect violations across the full privacy spectrum with statistical significance guarantees, which are supported by theory and simulations. Most importantly, and in contrast to prior work, our auditors do not require a user-specified sample size as an input. Rather, they adaptively determine a near-optimal number of samples needed to reach a decision, thereby avoiding the excessively large sample sizes common in many auditing studies. This reduction in sampling cost becomes especially beneficial for expensive training procedures such as DP-SGD. Our method supports both whitebox and blackbox settings and can also be executed in one-run frameworks.

1 Introduction

As data is becoming more valuable than gold, privacy is an increasingly pressing concern: how can we optimally utilize potentially sensitive data without compromising individuals' privacy? Differential Privacy (DP) [19] has emerged as a broadly accepted notion that hits a sweet spot between privacy and statistical utility. The idea behind DP is simple: The answer to a (privatized) query should not severely compromise the privacy of any individual record in the original dataset. A bit more formally, a mechanism satisfies DP if changing any single record has only a "small" effect on the outcome of the (privatized) query. Measuring the impact on outcomes naturally requires a notion of distance, which in classical DP is captured by two parameters ε and δ.

*Corresponding author: tim.kutta@math.au.dk

Following the initial study of (ε, δ)-DP—and fueled by the adoption of DP as an acceptable privacy notion by academia, industry and even governments [2–4, 16, 18, 21]—a number of DP variants have emerged as alternatives to tackle limitations of the (ε, δ)-DP notion. Examples include divergence-based relaxations of DP (e.g., Rényi-DP [36], zero-concentrated DP [12]), f-DP [17], and instance-based privacy concepts (e.g., DDP [7], PAC Advantage Privacy [43]).

Many initial results in DP considered feasibility, optimal utility and privacy tradeoffs, composability, and comparison between different privacy notions [14, 19, 20, 23, 25, 34, 39]. These results were typically either theoretical in nature—designing mechanisms and proving their privacy/utility—or they provided proof-of-concept implementations of such ideas and designs. The adoption of DP, however, for privacy-sensitive applications has put the notion under the scrutiny of the security research community, raising an additional question: How can we verify that (a given implementation of) a mechanism does not have a bug that makes it leak more than its specification?

DP Auditing The above question ignited a rich literature in security that aimed at testing/auditing whether implementations of common DP mechanisms satisfy their theoretical privacy guarantees. This led to a growing interest in empirical privacy auditing with results that range from using classical bug detection approaches—e.g., inspired by code obfuscation or worst-case instance discovery techniques—to deploying statistical and machine learning tools to compare neighboring distributions with respect to privacy-relevant events [9, 10, 33, 38].

The auditing toolbox is quite diverse, yet DP auditors can be broadly separated into two classes: *whitebox* auditors assume that the analyst knows specifics of the audited mechanism (e.g., parameters of the noising distribution), or even that full access to the concrete implementation is granted [9, 31, 37, 42]. In contrast, blackbox auditors only re-

quire blackbox (input/output) access to the DP mechanism.[1]

Naturally, the latter class poses more challenges, as tools from bug-finding methodologies cannot be used there and one needs to rely almost exclusively on statistical and machine learning-based methods as discussed below.

f-DP Auditing Despite the increasing literature on DP auditing, it is fair to say that most of this literature focuses on testing (implementations of) mechanisms for the classical (ε, δ)-DP notion. As discussed below, this puts several limitations, especially in the realm of blackbox and/or optimally efficient auditing. Notwithstanding, a number of recent works [6, 17, 33, 41] have demonstrated that *f-Differential Privacy*, in short, f-DP, can be more expressive than (ε, δ)-DP, while being more friendly to statistical methods for DP auditing. As such, f-DP has emerged as an excellent candidate for efficiently certifiable implementations in both whitebox and blackbox settings [37]. In fact, as discussed in [17] f-DP is a particularly suitable privacy notion for methods like *differentially private stochastic gradient descent* (in short, DP-SGD) that are increasingly popular in machine learning.

In a nutshell, f-DP connects the closeness of a (DP) mechanism's output distributions on neighboring datasets to the accuracy of (optimal) classifiers. If classifiers can reliably distinguish the output distributions, changes to the databases have a strong impact and privacy is low (we refer to Section 2.1 for a detailed definition). In classical statistics, the theory of optimal binary classifiers is well studied and well understood, providing a diverse toolbox for quantifying distinguishability between distributions. Thus it becomes evident that f-DP is far more amenable to tools from statistics and classical machine learning than (ε, δ)-DP.

Yet, from an auditing perspective the most crucial advantage of f-DP is the following: The notion of (ε, δ)-DP offers a very "localized" view of the algorithm's privacy as it does not make any statements about the potential trade-offs between ε and δ. Instead, f-DP provides a privacy notion that, intuitively, charts the entire (ε, δ) spectrum. This allows detection of privacy violations at a range of different scales simultaneously, and thereby yields much more powerful auditing procedures.

Sequential f-DP Auditing Motivated by the above, our work aims to push the boundaries of f-DP auditing both in the blackbox and the whitebox realm, by adapting powerful tools from statistics and machine learning. More concretely, we move from fixed-sample (fixed-batch) auditing procedures to a sequential auditor that adaptively determines how many samples are needed, with no major computational overhead. This approach promises substantially higher sample-efficiency and is supposed to complement existing auditing approaches by providing a generic sequential wrapper for any f-DP auditor. In the following, we outline the main ideas of our work and how they advance the state of the art.

Standard privacy auditors/tests take a fixed, predetermined number of outputs (from the mechanism), say n, and given these outputs run some statistical hypothesis test to assess privacy, e.g., [5, 29, 32]. Mathematically speaking, such methods can be proved to be optimal —by using optimal testing theory— but in practice they suffer from the "fixed sample size"-issue, which is a common bottleneck in testing: the ideal sample size (just large enough for detection, but not larger) of any statistical/hypothesis test is strongly case dependent (we refer to Section 2.2 for a detailed discussion). If the user chooses n too small, they risk inconclusive or false auditing results. If they choose n too large, they waste computational resources. Naturally, experiments and benchmarks err on the side of caution (to eliminate inconclusive or false results) and use extremely large sample sizes, often two orders of magnitude larger than necessary. Such oversampling may be harmless in foundational studies, but it naturally deters adoption, especially for auditing expensive algorithms such as DP-SGD.

Countering this trend toward ever more costly auditors, a recent strand of research aims to make DP auditing less expensive, among them [33, 37, 38], which focus on complex machine learning techniques. While their techniques already cut the computational cost dramatically, potentially down to training the algorithm only once, mostly they rely on evaluating canaries (see, e.g., [38]). As they note, while one would ideally score all n examples (i.e., set $m = n$ as number of canaries), doing so may be prohibitive and one often settles for $m < n$:

> *To get the strongest auditing results we would set $m = n$, but we usually want to set $m < n$. For example, computing the score of all n examples may be computationally prohibitive, so we only compute the scores of m examples.*

This gap suggests that sequential, anytime-valid procedures are useful even in the one-run setting: they can adaptively decide *how many* canaries to score without losing rigorous error control.

Finally, the very recent, parallel, and independent work [22] also made use of sequential tests for DP auditing. We only became aware of that work during the later stages of our paper. Notwithstanding, in the following we discuss the differences in scope and methodology: Their work from [22] focuses on the (ε, δ)-DP notion, which is distinct from our auditing of f-DP claims. This provides both qualitatively and quantitatively different results and we provide a detailed comparison in Section 4. In a nutshell, on the additive-noise benchmarks used by [22], our auditor rejects buggy mechanisms with substantially fewer samples in the small-ε regimes they report, using

[1] In some DP auditing work, blackbox still assumes the output distribution is known (e.g., audited mechanism is DP-SGD, hence corresponds to (subsampled) Gaussian tradeoff; likewise, LiRA-style attacks often model the distribution of losses as Gaussian [13, 37]. Our auditing methods rely on much weaker assumptions.

roughly 10%–25% of the paper-reported samples. Under a finite sample, it also continues to detect the buggy variants over a wider range of larger ε. This comparison, however, should not be read as a universal dominance guarantee: the two methods audit different privacy objects, rely on different statistical witnesses, and their sample efficiency also depends on the concrete construction instantiating the approaches.

We summarize the main contributions of this paper:

1) We provide the first sequential f-DP auditor.
2) We provide mathematical proofs for the performance of our auditor: In particular,
   a) we show that our auditor holds a user-determined significance level (false rejection rate) of $\gamma \in (0,1)$ (Theorem 3.1) ,
   b) and detects violations with minimal sample size, up to a logarithmic-factor (Theorem 3.1 and Remark 3.2).
3) Subroutines of our procedure involve a new tuning approach for optimal classifiers for DP auditing. Our new strategy even improves performance when used for fixed-batch f-DP auditing.
4) We demonstrate the effectiveness of our approach on synthetic and real-world experiments, including single run auditors.
5) We provide an open-source implementation.[2]

Paper structure. We briefly outline the structure of this work. In Section 2, we recall the key concepts underlying our methodology. Section 3 presents our main theoretical results, with the central contribution stated in Theorem 3.1. In Section 4, we experimentally investigate the performance of our approach: first using standard ground-truth settings, then comparing against the method of [22] using their favorable setup, and finally examining a real-world example in the spirit of [37] and [22]. Improvements, relative to the existing f-DP auditors in [5] are demonstrated in the Appendix. There, we also provide mathematical details and proofs for our new procedures.

2 Background

2.1 f-Differential Privacy

Differential Privacy (DP) is a concept to describe information leakage from a data processing mechanism $\mathcal{M}$. Qualitatively, privacy is high when for any pair of adjacent databases D, D' the output distributions $P \sim \mathcal{M}(D)$ and $Q \sim \mathcal{M}(D')$ are nearly indistinguishable. This "near indistinguishability" has been formalized in different ways (see e.g. [17, 19, 36] among others) and here we adopt the popular description of f-DP introduced in [17]. Intuitively, f-DP states that P, Q are hard to distinguish when there is no reliable way to classify whether data were generated by P or Q. Formally, suppose that upon making an observation X a user wants to decide which distribution was used to generate X, i.e.

$$X \sim P \qquad \text{versus} \qquad X \sim Q. \tag{1}$$

The decision is encoded by a binary classifier ϕ where $\phi(X) = 0$ means deciding that $X \sim P$ and $\phi(X) = 1$ means deciding that $X \sim Q$. No matter how good the classifier is, there is always room for error. Namely, one could wrongly classify $X \sim P$ as being generated by Q, and conversely $X \sim Q$ as being generated by P. We can express the misclassification probabilities as

$$\alpha_\phi := \mathbb{E}_{X\sim P}[\phi(X)] \quad \text{and} \quad \beta_\phi := 1 - \mathbb{E}_{X\sim Q}[\phi(X)]. \tag{2}$$

We can now assess the discrepancy between the distributions P and Q by virtue of the classification errors. Roughly speaking, if there exists a classifier ϕ that can distinguish data drawn from P versus Q with small error probabilities, then the distributions P and Q must differ substantially. Ideally, we would like to use the best possible classifier ϕ for this task. However, it turns out that there is no single best classifier that minimizes α_ϕ and β_ϕ simultaneously, because decreasing one error generally increases the other one. Rather, for any value $\alpha \in [0,1]$ there exists a best (possibly randomized) classifier ϕ^* with smallest attainable error β_{ϕ^*} under the side-constraint $\alpha_{\phi^*} \le \alpha$. This motivates the use of an entire *tradeoff function* that is a collection of all those pairs $(\alpha_{\phi^*}, \beta_{\phi^*})$.

Definition 2.1 (Tradeoff function)**.** For any two distributions P, Q defined on the same space, the tradeoff function is defined as

$$T(\alpha) := \inf\{\beta_\phi : \phi \text{ is a classifier with } \alpha_\phi \le \alpha\}, \ \alpha \in [0,1] \ .$$

T is a measure for the distance of output distributions P, Q. Accordingly, it can be used for privacy quantification. The corresponding notion of privacy is *f-DP*, where $f : [0,1] \to [0,1]$ is a convex and non-increasing function, with $f(0) = 1$ and $f(1) = 0$.

Definition 2.2 (f-DP)**.** A mechanism $\mathcal{M}$ satisfies f-DP if for every pair of neighboring datasets D, D', the corresponding tradeoff function T satisfies $T(\alpha) \ge f(\alpha)$ for all $\alpha \in [0,1]$.

2.2 Hypothesis testing and power analysis

Next, we consider the concept of statistical testing. Testing is related to classification in that a binary decision has to be reached based on observed data. Yet, there are two important

[2] https://github.com/martindunsche/sequential_fdp_auditing

differences: First, a decision is not based on a single observation X but rather on an entire sample $X_1, ..., X_n$. Second, the decision process involves a default belief, making the problem inherently asymmetric: one hypothesis is assumed to hold unless the data provides sufficient evidence to reject it. Abstractly, we can write down the two competing, mutually exclusive hypotheses as follows:

$$\begin{aligned} H_0 &: \text{the baseline model,} \\ H_1 &: \text{a collection of alternative models.} \end{aligned} \tag{3}$$

The baseline H_0 represents the default belief and is rejected in favor of H_1 only if observed data provide sufficiently strong evidence against it. For illustration, consider in the context of DP-auditing a postulated tradeoff curve f. Then, the null hypothesis H_0 states that a mechanism $\mathcal{M}$ satisfies f-DP, while H_1 states that it does not. Accordingly, if H_0 is rejected, we have high confidence that $\mathcal{M}$ does not satisfy f-DP.

As mentioned above, the statistical test reaches its decision between H_0 and H_1 based on a data sample $X_1, \ldots, X_n$. The ability to tell apart H_1 from H_0 depends on two main ingredients: First, the sample size n, where larger samples include more information (leading to a more informed decision) but are also more costly to collect. Second, the effect size, which is the deviation between the true data generating process and the model H_0. Recall that H_1 (in contrast to H_0) is not a single model - it is a collection of many possible models. Some of these models are similar to the baseline model H_0 and some are very different. Large discrepancies (large effect sizes) are naturally easier to identify than subtle ones (small effect sizes). In contrast to the sample size, the analyst has no influence on the effect size. Indeed, the effect size is unknown to them, since the true data generating process is unknown. This turns out to be a major problem in hypothesis testing.

The performance of a statistical test is captured by two fundamental probabilities:

$$\begin{aligned} \text{significance level (false rejection rate): } & \mathbb{P}(\text{reject } H_0 \mid H_0), \\ \text{power (correct rejection rate): } & \mathbb{P}(\text{reject } H_0 \mid H_1). \end{aligned} \tag{4}$$

Across empirical sciences, it is common to fix the significance level at some low value, typically 5%. Under this constraint, the main aim becomes attaining a certain high level of power (say 80%). Reaching a desired power is non-trivial because power is interconnected with both the sample size and the effect size; see Figure 1. More precisely, larger samples and larger effect sizes both lead to more power. In an idealized setting where the effect size were known to the analyst, they could analytically determine the minimal required sample size, denoted by $n_{\min}$, to attain a desired power. However, in reality the effect size is unknown. To hedge against this uncertainty, analysts often adopt a conservative strategy and employ substantially larger samples than theoretically necessary, such that $n \gg n_{\min}$.

In DP auditing, it is not uncommon to encounter sample sizes that exceed the theoretical minimum by two to three orders of magnitude, placing a big computational burden on the analyst. The impossibility of selecting the optimal sample size before testing motivates an adaptive choice that is made during testing. This strategy leads to close-to-optimal results and is the subject of sequential testing.

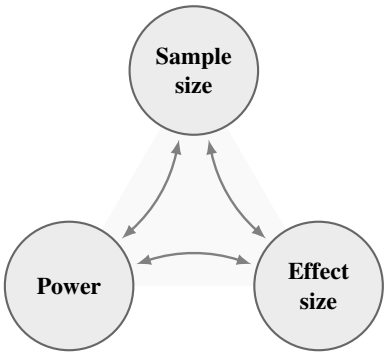


Figure 1: Power, sample size, and effect size form a triad in which any two quantities determine the third. Thus, an optimal sample size can only be computed once both the desired power and effect size are specified.

2.3 Sequential testing

Consider the setup from the previous section. To avoid the use of excessively large samples for hypothesis testing, sequential procedures have been developed that adaptively minimize the sample size during the testing process. The core idea is simple: In practice, an analyst rarely collects all data at once; rather, observations arrive sequentially. First, X_1 is observed, then, (X_1, X_2), and so on. By monitoring the accumulating data in real time, the analyst may already obtain sufficient evidence to reject H_0 in favor of H_1 after observing only a small prefix $(X_1, \ldots, X_k)$. Here, the time point k at which H_0 is first rejected is a random variable, but it typically concentrates near the theoretical minimum $n_{\min}$.

Designing valid sequential procedures is non-trivial. An ad-hoc approach might be to repeatedly apply a standard fixed-sample test, every time a new observation is made. Unfortunately, this naive strategy cannot work, because the false rejection rate grows uncontrollably. We illustrate this point shortly with the next example.

Example 2.3. Consider data from a normal distribution $X_i \sim \mathcal{N}(\mu, 1)$, with mean $\mu \in \mathbb{R}$. The hypotheses under investigation are $H_0 : \mu = 0$ vs. $H_1 : \mu \neq 0$. To sequentially check H_0 vs. H_1, the analyst runs a t-test repeatedly whenever a new observation is made and stops after the first rejection. Each t-test individually has a significance level of 5%. Now, let us assume that the true data are indeed generated by the

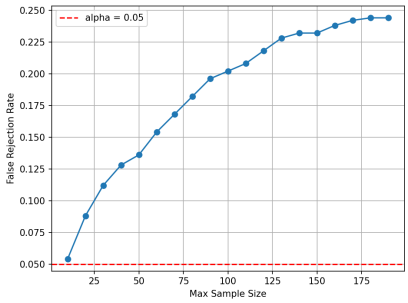


Figure 2: False rejection rates in sequential testing without proper adjustment.

model $\mathcal{N}(0,1)$ and thus, any rejection of H_0 would be erroneous. We have run $1,000$ simulations to obtain empirical false rejection rates and plotted them in Figure 2. As we can see, rather than holding the desired significance level (red horizontal line), the false rejection rate rises rapidly and is already $\approx 24\%$ after the first 200 datapoints. We do not display it here, but it can be mathematically proven that the false rejection rate increases ever further, converging to 100% if testing goes on indefinitely. This means a false rejection is eventually guaranteed with the naive procedure.

The above example illustrates the need for designated sequential tests to ensure that the significance level holds, even when testing goes on for a very long time. Rather than consisting of a fixed-sample test that is applied over and over again, sequential tests compute an evolving statistic that can be approximated by a stochastic process. The probability of that process crossing a boundary, which is analytically tractable, then gives a closed-form for the false rejection rate of the sequential test. The development of such procedures is mathematically challenging and relies on modern insights from open-ended Gaussian approximations such as developed by [8].

3 Methodology

In this section, we present a sequential f-DP auditing approach that adapts the sample size (number of algorithmic runs) to the unknown level of privacy violation. Strong violations can be detected with a few samples, while subtler effects automatically trigger more sampling.

3.1 Sequential privacy auditing

We fix a pair of adjacent databases D, D' and denote the corresponding output distributions of the mechanism $\mathcal{M}$ by $P \sim \mathcal{M}(D)$ and $Q \sim \mathcal{M}(D')$. Recall the central Definitions 2.1 and 2.2. In Definition 2.1, we noticed that the discrepancy between P and Q is captured by the tradeoff function T. In Definition 2.2, we said that f-DP holds if the function T lies geometrically above the function f. The task of an f-DP auditor is then to check for a candidate function f, whether f-DP holds. This task is naturally framed as a hypothesis test (see Section 2.2), where we decide between the competing hypotheses of f-DP (H_0) and no f-DP (H_1)

$$\begin{aligned} H_0 &: T(\alpha) \geq f(\alpha) \quad \forall \alpha \in [0,1], \\ H_1 &: \exists\, \alpha^* \in [0,1] \text{ such that } T(\alpha^*) < f(\alpha^*). \end{aligned} \tag{5}$$

To decide between H_0 and H_1, the analyst may include prior knowledge in their decision process. In an ideal whitebox scenario, the analyst already knows a powerful classifier ϕ to distinguish P and Q (see Section 2.1), while in a blackbox setting only minimal knowledge is available, e.g. that P and Q have probability densities. Our methodology is compatible with both settings and details are discussed below. In either case, the analyst bases their final verdict on samples of algorithmic outputs. More precisely, they draw independently $X_1, X_2, \ldots \sim P$ and $Y_1, Y_2, \ldots \sim Q$ and employ a statistical test for the hypotheses pair (5). This test should have the two properties described after Eq. (4): First, it should bound the false rejection rate by a small, user-determined number $\gamma \in (0,1)$, the significance level. Second, when f-DP is violated, the test should reject H_0 using as few samples of algorithmic outputs as possible. We therefore use a sequential testing procedure (background in Section 2.3) that we describe next.

3.2 Sequential auditing scheme

In the following, we verbally describe our new sequential auditing scheme, termed *Advanced Privacy Testing,* in short APT (see Algorithm 1 for a blueprint of APT). For convenience, we use the line numbering in Algorithm 1 to refer to specific steps, following the notation $\langle APT.L.i \rangle$ (Algorithm 1, line number i).

Prior to the testing phase, APT generates a small burn-in sample $(X_1, Y_1), \ldots, (X_M, Y_M)$, which is used for parameter calibration $\langle APT.L.2\text{-}3 \rangle$. This sample is later reused during the testing phase $\langle APT.L.12 \rangle$. Next, APT constructs a classifier ϕ, using the burn-in sample and (if available) prior knowledge of the analyst $\langle APT.L.3 \rangle$. Note that there are very different ways to instantiate the classifier in the APT blueprint and the concrete scenario dictates what a good choice is. For instance, in a whitebox scenario, the analyst might know an optimal classifier ϕ and may input it to APT without reference to the data. On the opposite end of the spectrum, in a blackbox scenario, an initial classifier may need to be constructed from scratch based on the burn-in sample, and may be refined as additional data become available. We refer to the following sections for the concrete classifier choices that instantiate the APT blueprint in our results.

After ϕ has been selected, the main task becomes approximation of its classification errors $(\alpha_\phi, \beta_\phi)$ $\langle APT.L.4, 12 \rangle$ defined in (2). Recall that, by definition, the pair $(\alpha_\phi, \beta_\phi)$ lies on

Algorithm 1 APT: Advanced Privacy Testing

Require: Distribution P, Q Burn-in size M, bandwidth h, candidate vector η, refresh period M_1, evaluation period M_2, claimed curve f, critical $1-\gamma/2$ value $q_{1-\gamma/2}$, max iterations k_{max}.

Ensure: Online monitoring for violations of a claimed privacy curve.

function SEQUENTIAL-AUDIT($M, h, \eta, M_1, M_2, f, q_{1-\gamma/2}, k_{max}$)
  Draw M samples from P and Q .
  Define classifier

$$\phi(\cdot) \leftarrow \text{BUILDCLASSIFIER}(\{X_i\}_{i=1}^M, \{Y_i\}_{i=1}^M, h, \eta, f) \ .$$

  Compute initial statistics:

$$\hat{\alpha}_\phi := \frac{1}{M}\sum_{i=1}^{M} \phi(X_i), \quad \hat{\beta}_\phi := 1 - \frac{1}{M}\sum_{i=1}^{M} \phi(Y_i).$$

  Compute confidence-adjusted estimators:

$$\hat{T}_\alpha, \hat{T}_\beta \leftarrow \text{CONFADJ}(\hat{\alpha}_\phi, \hat{\beta}_\phi, q_{1-\gamma/2}, M). \qquad \triangleright \text{ Algorithm 4}$$

  for $k = M+1, M+2, \ldots, k_{max}$ **do**
    Draw $X_k \sim P$, $Y_k \sim Q$.
    if $k \bmod M_1 = 0$ **then**
      Update $\phi(\cdot)$.
    end if
    if $k \bmod M_2 = 0$ **then**
      Update $\hat{\alpha}_\phi, \hat{\beta}_\phi$ using $\{X_i, Y_i\}_{i \le k}$.
      Update $\hat{T}_\alpha, \hat{T}_\beta \leftarrow \text{CONFADJ}(\hat{\alpha}_\phi, \hat{\beta}_\phi, q_{1-\gamma/2}, M)$.
      if $\hat{T}_\beta < f(\hat{T}_\alpha)$ **then**
        return "`DP violation`" at time k.
      else
        Continue monitoring.
      end if
    end if
  end for
end function

or above the tradeoff curve T. Thus if APT's measurements indicate that $(\alpha_\phi, \beta_\phi)$ is below f, we interpret this as a signal that T is also likely below f and thus f-DP is violated. Note, however, that due to randomness in our samples, the quantities $(\hat{\alpha}_\phi, \hat{\beta}_\phi)$ are noisy estimators of $(\alpha_\phi, \beta_\phi)$; therefore, to control false rejections, we reject f-DP only when $(\hat{\alpha}_\phi, \hat{\beta}_\phi)$ lies below f by a sufficiently large margin $\langle APT.L.14-15\rangle$—the precise amount is determined by Algorithm 4 $\langle APT.L.5, 13\rangle$. If no f-DP violation is detected, the algorithm keeps running $\langle APT.L.17\rangle$ until a maximum number of iterations $k_{\max}$ is reached. In our theory, we may set $k_{\max} = \infty$ (arbitrarily long DP-auditing) and still hold statistical significance guarantees.

The mathematical crux of our approach lies in the confidence adjustment by Algorithm 4 that allows the control of the false rejection rate. To obtain confidence regions for the empirical classification error, it is standard to use a Hoeffding bound, as e.g. done in [5, 32]. However, these confidence bounds are of fixed-batch type and therefore inappropriate for the more stringent sequential testing setup. Here sequential decision boundaries are needed, which have recently become popular in the literature under the label of "anytime valid inference" (see [24] for an exposition). Our Algorithm 4 compares empirical classification errors, as more and more data arrive, to the fluctuations of a Brownian motion. This comparison is based on state-of-the-art tools from Gaussian approximation theory, developed by [8]. Some background on the Brownian motion can be found in the Appendix. For the Brownian motion, one can derive boundary functions that it only crosses with a probability of γ - exactly corresponding to our significance level. Using the same boundary functions in Algorithm 4 to describe the fluctuations of our empirical classification errors yields a valid sequential inference procedure.

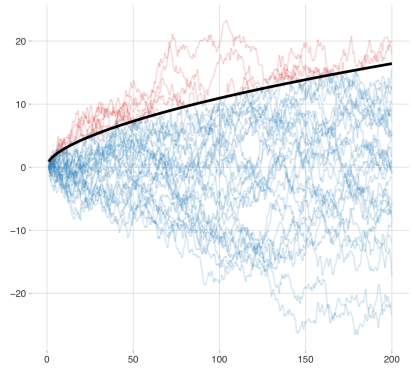


Figure 3: Realizations of the standard Brownian motion on the interval $[1, 200]$ compared to a boundary function $g(x) = \sqrt{(x/4)\log(20+x)}$ shown in solid black. Blue paths stay below the boundary function and red paths cross the boundary at some point.

Below, in Theorem 3.1, we provide a formal performance guarantee for our approach. The theoretical result is formulated here for one fixed classifier ϕ, i.e. without updating, a situation most typical in a white-box setting. An advantage of this presentation is its relative simplicity compared to scenarios with updating rules. The proof, presented in the Appendix, Section A.1, is based on classical open-ended Gaussian approximations that are available for i.i.d. random variables. The case of updating rules is more complicated, because it introduces serial dependence. For instance, suppose that at time k the classifier $\hat{\phi}$ draws on previous data $X_1, \ldots, X_{k-1}$; then the evaluations $\hat{\phi}(X_i), \hat{\phi}(X_j)$ for two indices $i \neq j$ are no longer independent, and the distribution of outputs may evolve over time with $\hat{\phi}$. In the Appendix, Section C, we present theory for classifiers that evolve over time. The main assumption is that the sequence of classification outcomes remains sufficiently regular for Gaussian approximations to apply (now for weakly dependent data). Practically, this points toward update rules that stabilize over time, so that historical samples

have only limited influence on the classifier at later stages. Some examples of such classifiers are discussed in the next section. The assumption of weak dependence is formalized using classical concepts from time series analysis such as strong mixing. The justification for adaptive classifiers, also given in Section C, is then much more involved, using recent advances in Prokhorov-metric approximation for partial sum processes of dependent, non-stationary data. We also discuss technical assumptions there in more detail.

For now, let us focus on the exemplary case of one fixed ϕ, which is the subject of Theorem 3.1 below. The theorem has three statements. The first two, (a)-(b), involve $o(1)$ terms, which are negligible for larger M. In practice, moderate values such as $M = 50$ are sufficient for a tight approximation. The parameter $\gamma \in (0,1)$ is the user-determined significance level and practically fixed at some small value, such as 5% (see Section 2.2). According to parts (a) and (b), γ bounds the risk of a false rejection if f-DP actually holds. Smaller values of γ ensure fewer false rejections, but come at the cost of slower detection of actual privacy violations. Part (c) holds for any M and states that a privacy violation will be detected eventually (for large enough k) with certainty.

Theorem 3.1. Let $\phi(\cdot)$ be a classifier and $\gamma \in (0,1)$ a significance level. Then the following statements hold:

a) If $(\alpha_\phi, \beta_\phi) \in (0,1)^2$ satisfies $f(\alpha_\phi) < \beta_\phi$ (DP claim satisfied by a positive margin), it follows that

$$\mathbb{P}\big(\text{APT outputs for some } k \texttt{ "DP violation"}\big) = o(1).$$

b) If $(\alpha_\phi, \beta_\phi) \in (0,1)^2$ satisfies $f(\alpha_\phi) = \beta_\phi$ (DP claim exactly satisfied), it follows that

$$\mathbb{P}\big(\text{APT outputs for some } k \texttt{ "DP violation"}\big) = \gamma + o(1).$$

c) If $(\alpha_\phi, \beta_\phi) \in (0,1)^2$ satisfies $f(\alpha_\phi) > \beta_\phi$ (DP claim violated), it follows that

$$\mathbb{P}\big(\text{APT outputs for some } k \texttt{ "DP violation"}\big) = 1.$$

Remark 3.2. Theorem 3.1 implies that a violation of $f-$DP is detected by the APT algorithm eventually (i.e. if samples are large enough). In the following we discuss how large the samples required by APT are, compared to the theoretical optimum – a fixed-batch test with optimal sample size. Recall from our above discussion that using such an optimal test is practically impossible, because the effect size is unknown. In summary, we see that the samples used by APT are within a logarithmic factor of the theoretical optimum.

Let us now suppose that $f-$DP is violated and that we have a candidate classifier ϕ, with $(\alpha_\phi, \beta_\phi)$ below the claimed curve f. The effect size (Section 2.2) is the distance between $(\alpha_\phi, \beta_\phi)$ and f, and we denote it here by δ. An ideal test would now be based on a confidence region around $(\alpha_\phi, \beta_\phi)$, e.g. using a Hoeffding bound (or t-test) with sample size dependent on δ. Some simple calculations show that the minimal required sample size to detect the violation would have to be of size $O(\delta^{-2})$ as δ becomes small. Our sequential approach yields similar values. Indeed, depending on the relation of δ and M, it follows that the sample size in the sequential testing is at best $O_P(\delta^{-2})$ (equal to minimum up to a constant) and at worst $O_P(\delta^{-2}\log(\delta^{-1}))$ (within a log-factor of the minimum). We thus see that there is only a minimal sampling overhead of sequential testing compared to the (infeasible) mathematically optimal procedure.

3.3 Our classifiers suite

We next describe how we instantiate the APT blueprint with (close to) optimal classifiers in different relevant scenarios. In a nutshell, our classifiers adapt the Neyman-Pearson Lemma from classical statistics (*cf.* [30, p.60]) to the different scenarios. More concretely: suppose that P and Q have probability densities, denoted by p and q, respectively. The Neyman-Pearson Lemma defines the Likelihood Ratio approach: For constants $\lambda \in [0,1]$ and $\eta > 0$, and a random bit $B \sim \text{Ber}(\lambda)$, it takes the form

$$\phi_{\eta,\lambda}(x) := \begin{cases} 0, & q(x)/p(x) < \eta \\ 0, & q(x)/p(x) = \eta \ \&\ B = 0 \\ 1, & \text{else.} \end{cases} \tag{6}$$

Adapting the lemma to the f-DP formulation and terminology it states that any ϕ of the above form is optimal, in the sense that the error pair $(\alpha_\phi, \beta_\phi)$ lies exactly on the trade-off curve T. For f-DP auditing, it would therefore suffice to consider classifiers from the set $\{\phi_{\eta,\lambda}\}_{\eta,\lambda}$. Obtaining such optimal classifiers is different in blackbox and whitebox settings, and we discuss both cases below:

Blackbox Approach In a blackbox setting, employing optimal classifiers faces two main obstacles. First, the densities p and q are unknown, which means that the corresponding optimal classifiers $\phi_{\eta,\lambda}$ in (6) are unknown as well. We obtain feasible substitutes for p and q by estimation - in our experiments, we employ kernel density estimation although other approaches (for instance, neural network based estimators) are equally suitable. Density estimates are updated sequentially as more data become available.

Second, even if the entire family $\{\phi_{\eta,\lambda}\}_{\eta,\lambda}$ were somehow available, one would still need to decide which specific pair (η,λ) to use. For most models, η is the decisive parameter. For example, in the Gaussian case discussed below, the choice of λ is entirely irrelevant. Therefore, we focus here on η exclusively, setting $\lambda = 0$ and suppressing dependence on λ in our notation with $\phi_\eta := \phi_{\eta,0}$. Adaptations of our method that include $\lambda \neq 0$ are probably possible, but they seem to be less relevant for DP-auditing. Each value of η can be associated with a unique operating point $(\alpha_{\phi_\eta}, \beta_{\phi_\eta})$, and in Section 3.4

below, we discuss a simple geometric strategy to choose the optimal η.

The corresponding methods are summarized below in Algorithm 3 (estimation of T), Algorithm 6 (selection of optimal η) and can then be integrated as subroutines into the APT Algorithm 1.

Whitebox Approach If the analyst has additional prior knowledge about the distributions P and Q, we may depart from the non-parametric blackbox methods above and instead adopt a parametric model that offers higher sample efficiency. A classical assumption is that the outputs under P and Q are (approximately) Gaussian with a common variance σ^2 and different means μ_P and μ_Q (see e.g. [13]). Under this model, the Likelihood Ratio classifier is equivalent to a simple threshold classifier on the scalar output s: $\phi_\eta(s) = \mathbf{1}\{s \geq \eta\}$, namely, we decide P if $s \geq \eta$ and Q otherwise. The appeal of this approach is that, under the Gaussian equal-variance model, the optimal Likelihood Ratio classifier depends only on the two means (after standardization by the common variance). Moreover, even when the true output distributions deviate moderately from normality, the Gaussian plug-in Likelihood Ratio classifier often remains a powerful tool for auditing purposes. As before, we can use Algorithm 6 to detect a critical η.

3.4 Likelihood Ratio tuning

In the previous section, we have focused on approximating density ratios $p(x)/q(x)$ (either non-parametric or parametric), used for the optimal classifier from the Neyman-Pearson Lemma. We will now focus on the choice of the optimal threshold η^* in Eq. (6). Recall that any η implies a Likelihood Ratio classifier ϕ_η, and to emphasize our focus on η, define

$$\alpha_{\phi_\eta} =: \alpha(\eta), \qquad \beta_{\phi_\eta} =: \beta(\eta).$$

Our aim is to select a classifier that maximizes our power against privacy violations. A first intuition for this is to select a pair $(\alpha(\eta), \beta(\eta))$ that lies as far as possible under the claimed privacy curve f. This intuition was used by [5], where the selection rule

$$\eta^* := \arg\max_\eta f(\hat{\alpha}(\eta)) - \hat{\beta}(\hat{\alpha}(\eta)) \,, \tag{7}$$

was suggested, where $\hat{\beta}(\cdot)$ is an estimator of the true unknown tradeoff curve. In (7), η^* is selected so that $(\alpha(\eta), \beta(\eta))$ lies as far as possible under f in vertical direction. In the following, we use a modification of this rule that significantly improves the detection of DP violations.

For our modification to make sense, notice that detecting DP violations does not directly depend on the one-dimensional, vertical distance between $(\alpha(\eta), \beta(\eta))$ and f. Rather, it depends on a two-dimensional quantity, and a simple proxy is the largest axis-aligned square around the center

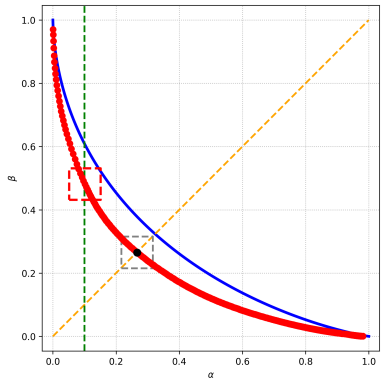


Figure 4: The blue line corresponds to f, while the red line represents the estimated tradeoff curve $\hat{T}$. The vertical green line and diagonal yellow line correspond to the rules of selecting η in (7) and (8), respectively. The rule (8) maximizes the size of the square that can be fit below f – the central quantity for detecting DP-violations.

$(\alpha(\eta), \beta(\eta))$, that still fits below f. The upper-right corner of this square lies on the 45° direction, from $(\alpha(\eta), \beta(\eta))$. This motivates selecting a point that is well separated from f along this diagonal direction, rather than only in the vertical direction as in (7). To make this geometric idea mathematically precise, consider for any estimate $(\hat{\alpha}(\eta), \hat{\beta}(\eta))$ the line

$$\beta(\alpha) = \hat{\beta}(\eta) + \big(\alpha - \hat{\alpha}(\eta)\big) \,,$$

which passes through $(\hat{\alpha}(\eta), \hat{\beta}(\eta))$ with slope 1. We can find the intersecting point of this line and f by solving the equation

$$f(\alpha) = \hat{\beta}(\eta) + \big(\alpha - \hat{\alpha}(\eta)\big)$$

for α. Proceeding this way, we can for each η compute the Euclidean distance between $(\hat{\alpha}(\eta), \hat{\beta}(\eta))$ and its intersection point $(\alpha_{\text{int}}(\eta), f(\alpha_{\text{int}}(\eta)))$. The critical η^* is now the following:

$$\eta^* := \arg\max_\eta \|(\hat{\alpha}(\eta), \hat{\beta}(\eta)) - (\alpha_{\text{int}}(\eta), f(\alpha_{\text{int}}(\eta)))\|_2 \,. \tag{8}$$

In Figure 4 we illustrate our general idea. Our new approach selects a point that maximizes the size of the square under the tradeoff curve (which is directly linked with more power to detect DP violations). In contrast, the method by [5] selects a less appropriate point, and we see that the same square now intersects with the curve f (it does not fit under it anymore).

4 Experiments

In this section, we evaluate the proposed sequential auditor. We first consider controlled ground-truth benchmarks in both blackbox and whitebox settings. In the blackbox setting, the analyst observes only mechanism outputs, whereas in the

whitebox setting, the analyst additionally has prior knowledge about the mechanism, such as the class of noising distributions. We then compare our method with the sequential (ε,δ)-DP auditor proposed by [22]. Finally, we demonstrate in a realistic one-run DP-SGD canary audit, how our sequential pipeline can reduce the number of canary scores needed for auditing. We now formulate the common setup for the ground-truth experiments below.

Setup for controlled ground-truth experiments. Our experimental framework follows the well-established and accepted methodology for estimating and auditing various notions of differential privacy (see e.g. [5, 29]): Throughout, we consider neighboring databases $D, D' \in [0,1]^r$ with $r = 10$ records. A standard choice in the literature, which maximizes the ℓ_1-distance while preserving adjacency, is

$$D = (0,\dots,0), \qquad D' = (1,0,\dots,0), \tag{9}$$

though similar behavior is observed for other adjacent pairs. For the choice of databases, we also refer to [29]. We denote by $S(x) = \sum_{i=1}^{10} x_i$ the summary statistic used in our additive-noise experiments.

Mechanisms We evaluate three mechanisms commonly studied in privacy research: the Gaussian mechanism, the Laplace mechanism, and DP-SGD. These mechanisms produce quite different output distributions and therefore provide a meaningful benchmark. Below we briefly describe each mechanism together with the parameter settings used in our experiments. We also provide, for each mechanism, a mathematically closed form for the tradeoff curve, that can serve as ground truth in our below experiments.

Additive Noise Mechanisms For both the Gaussian and Laplace mechanisms, the released value takes the form

$$\mathcal{M}(x) = S(x) + Y,$$

where $Y \sim \mathrm{Lap}(0,b)$ or $Y \sim \mathcal{N}(0,\sigma^2)$. When the constants b, σ are appropriately calibrated, both mechanisms satisfy f-DP.

For the Gaussian mechanism, we fix $\sigma = 1$. Following [17], the corresponding tradeoff curve is then given by the closed form

$$T_{\mathrm{Gauss}}(\alpha) = \Phi\big(\Phi^{-1}(1-\alpha) - \mu\big),$$

where Φ is the cumulative distribution function of the standard normal. We notice that $T_{\mathrm{Gauss}}(\alpha)$ corresponds to the tradeoff curve between $\mathcal{N}(0,1)$ and $\mathcal{N}(\mu,1)$.

For the Laplace mechanism, we fix $b = 1$ and use another closed form derived by [17], namely

$$T_{\mathrm{Lap}}(\alpha) = \begin{cases} 1 - e^{\mu}\alpha, & \alpha < e^{-\mu}/2, \\ e^{-\mu}/(4\alpha), & e^{-\mu}/2 \le \alpha \le 1/2, \\ e^{-\mu}(1-\alpha), & \alpha > 1/2. \end{cases}$$

DP-SGD Deriving a closed-form expression for the tradeoff curve of DP-SGD is in general challenging and often infeasible. However, in a special case, [5] have derived a closed form that we can use as ground truth for our validation.

First, recall that DP-SGD aims to privately approximate the solution to the empirical minimization problem

$$\theta^* = \arg\min_{\theta\in\Theta} \mathcal{L}_x(\theta), \qquad \mathcal{L}_x(\theta) = \frac{1}{r}\sum_{i=1}^{r} \ell(\theta, x_i).$$

In our experiments, we choose $\ell(\theta, x_i) = \frac{1}{2}(\theta - x_i)^2$, initialize $\theta_0 = 0$, and set $\Theta = \mathbb{R}$. The remaining hyperparameters are fixed to $\sigma = 0.2$, $\rho = 0.2$, $\tau = 10$, and $m = 5$. Then, according to [5], the tradeoff curve is given by

$$T_{\mathrm{SGD}}(\alpha) = \sum_{I\subset\{1,\dots,\tau\}} \frac{1}{2^\tau} \Phi\Big(\Phi^{-1}(1-\alpha) - \frac{\mu_I}{\bar{\sigma}}\Big),$$

where

$$\mu_I := \sum_{t\in I} \frac{\rho(1-\rho)^{\tau-t}}{m}, \qquad \bar{\sigma}^2 = \rho^2\sigma^2 \frac{1-(1-\rho)^{2\tau}}{1-(1-\rho)^2}. \tag{10}$$

Experimental Parameters We use the same global parameters across all mechanisms: the refresh period M_1 is chosen adaptively (see Algorithm 7 with threshold 0.1), evaluation period $M_2 = 10$ is static, maximum number of iterations $k_{\max} = 10{,}000$, and significance level $\gamma = 0.05$. All empirical rejection probabilities are derived using $1{,}000$ repetitions.

Gaussian We introduce privacy violations by miscalibrating the privacy parameter μ. Data are generated under $\mu = 1$ and we audit for the claimed curves when $\mu = 0.5,\ 0.8,\ 1$.

Laplace We use the same strategy as for the Gaussian mechanism. The true calibration uses $\mu = 1$, while we audit under the claims $\mu = 0.5,\ 0.8,\ 1$.

DP-SGD For DP-SGD, instead of μ, we vary the number of iterations τ. The true mechanism runs with $\tau = 10$, while we audit under the claims $\tau = 5,\ 7,\ 10$.

Summing up, for each mechanism we thus consider three scenarios: *strong violation*, *subtle violation*, and *no violation*.

4.1 Blackbox

We now evaluate the three mechanisms under the blackbox setting. Before proceeding, we briefly recall the construction of the classifier. Algorithm 3 describes the perturbed Likelihood Ratio estimator introduced in [5], which approximates the optimal Likelihood Ratio classifier. In addition, we derive the critical decision rule g required by the sequential auditing procedure in Algorithm 1. The complete blackbox auditing pipeline is summarized in Algorithm 1 and by setting (cf. Algorithm 2)

$$\mathcal{L}(\{X_i\}_{i=1}^m, \{Y_i\}_{i=1}^m) = \log(\hat{p}) - \log(\hat{q}),$$

where $\hat{q}$ and $\hat{p}$ are KDEs for the unknown densities p, q.

Blackbox Parameters For the KDE-based blackbox procedure, we introduce a few additional parameters. We set the maximum threshold value to $\eta_{\max} = 15$ (i.e. $\log(\eta_{\max}) \approx 2.7$) and the perturbation level in Algorithm 3 to $h = 0.1$. Bandwidths for the KDEs are selected in a data-adaptive manner using the package [40].

4.2 Whitebox

In the whitebox setting we focus on the Gaussian mechanism. For simplicity, we fix $\sigma = 1$. We instantiate BUILDCLASSIFIER with a parametric learning rule $\mathcal{L}_{\text{Gauss}}$ that fits the means

$$\hat{\mu}_P := \frac{1}{m}\sum_{i=1}^{m} X_i, \qquad \hat{\mu}_Q := \frac{1}{m}\sum_{i=1}^{m} Y_i,$$

and uses the analytic Gaussian tradeoff. We consider an equidistant threshold grid $\mathcal{T} \subset [\min(X,Y), \max(X,Y)]$. For each $\eta \in \mathcal{T}$, the induced classifier is the threshold rule

$$\phi_\eta(x) := \mathbf{1}\{x \geq \eta\}.$$

Moreover, the corresponding estimated tradeoff point is available in closed form

$$\hat{\alpha}(\eta) = 1 - \Phi(\eta - \hat{\mu}_P), \qquad \hat{\beta}(\eta) = \Phi(\eta - \hat{\mu}_Q).$$

Finally, BUILDCLASSIFIER selects η^* by applying (8) to the curve $\{(\hat{\alpha}(\eta), \hat{\beta}(\eta)) : \eta \in \mathcal{T}\}$ and returns $\phi(\cdot) = \phi_{\eta^*}(\cdot)$.

To illustrate the impact of sequential auditing, we also compare our *sequential* APT procedure to a *fixed-batch* (offline) baseline and a previous auditor which also operates in *fixed-batch* (see full version for more details).

Interpretation of the results Each curve summarizes 1,000 independent repetitions of the auditing experiment. For a given stopping time n on the horizontal axis, we record whether the procedure has rejected by time n, and plot the resulting empirical rejection rate on the vertical axis. Thus, the blue curve can be interpreted as a power-over-time curve. Essentially it captures how quickly the auditor accumulates evidence and starts rejecting as more samples are observed. For each fixed value of n, the empirical rejection rate is itself random. If the true rejection probability is p_n, then the number of rejections over R repetitions follows a binomial distribution $\text{Bin}(R, p_n)$, and the empirical rejection rate is this count divided by R. To visualize the resulting uncertainty, we have included shaded ribbons showing pointwise 95% confidence intervals for the rejection probabilities. These intervals are very narrow in our experiments. The red dashed horizontal line indicates the significance level $\gamma = 0.05$. Under the null hypothesis (third column), the blue curve should stay near this line, while under alternatives (first and second column) it should rise above it.

Across all simulation settings, the sequential auditor behaves as intended under H_0: the empirical rejection probability remains at (or below) the prescribed level γ. Here, we do not expect to fully use the prescribed false positive rate γ, as the procedure is supposed to hold even when our sample size becomes arbitrarily large, i.e. $n \to \infty$. Under different alternatives, the procedure detects the signal quickly, often after only a few hundred samples. Consequently, only minor effort is needed in many cases. Stronger privacy violations (larger effect size) impact the detection rate positively, which is a classical and expected phenomenon.

4.3 Comparison to Prior Work

An empirical comparison with existing work is only meaningful for other sequential methods. The only existing work in this direction seems to be the (independent and parallel) method by [22] for (ε, δ)-DP. A direct apples-to-apples comparison is inherently difficult, because the two works audit different privacy objects. We test a claimed f-DP tradeoff curve: the null is that the hypothesized tradeoff function T for neighboring output distributions P and Q satisfies that

$$H_0 : T(\alpha) \geq f(\alpha), \quad \forall \alpha \in [0,1], \qquad \text{(c.f.(5))}$$

whereas rejection means that a classifier-induced pair $(\alpha_\phi, \beta_\phi)$ provides evidence that the claimed tradeoff curve is violated.

By contrast, [22] audit approximate DP. Since the hockey-stick divergence defining (ε, δ)-DP is difficult to test directly in a black-box setting, their method tests an MMD surrogate:

$$H_0^{\text{MMD}} : \quad \text{MMD}(P,Q) \leq \tau_{\text{MMD}}(\varepsilon, \delta), \qquad (11)$$

where the constant $\tau_{\text{MMD}}(\varepsilon, \delta)$ is defined as $\sqrt{2}\left(1 - \frac{2(1-\delta)}{1+e^\varepsilon}\right)$. Thus $\text{MMD}(P,Q) > \tau_{\text{MMD}}(\varepsilon, \delta)$ indicates a violation of the claimed (ε, δ)-DP guarantee. We note that, however, the converse does not hold: a mechanism may violate approximate DP while still having small MMD.

Because the two methods differ in their testing objectives, privacy notions, and underlying statistical objects, we do not aim to establish a universal sample-efficiency dominance claim. Indeed, the sample efficiency of [22] is governed by the MMD margin

$$\Delta_{\text{MMD}} := \text{MMD}(P,Q) - \tau_{\text{MMD}}(\varepsilon, \delta),$$

which depends on the choice of kernel and bandwidth for the MMD test-function class, the neighboring output distributions P, Q, and the claimed privacy level. In contrast, the sample efficiency of our auditor is governed by the separation between the classifier-induced error pair and the claimed tradeoff curve, which depends on the construction of the classifier and the neighboring output distributions P, Q.

Thus, favorable regimes in terms of sample-efficiency may differ. We do, however, want to point out that for [22], the

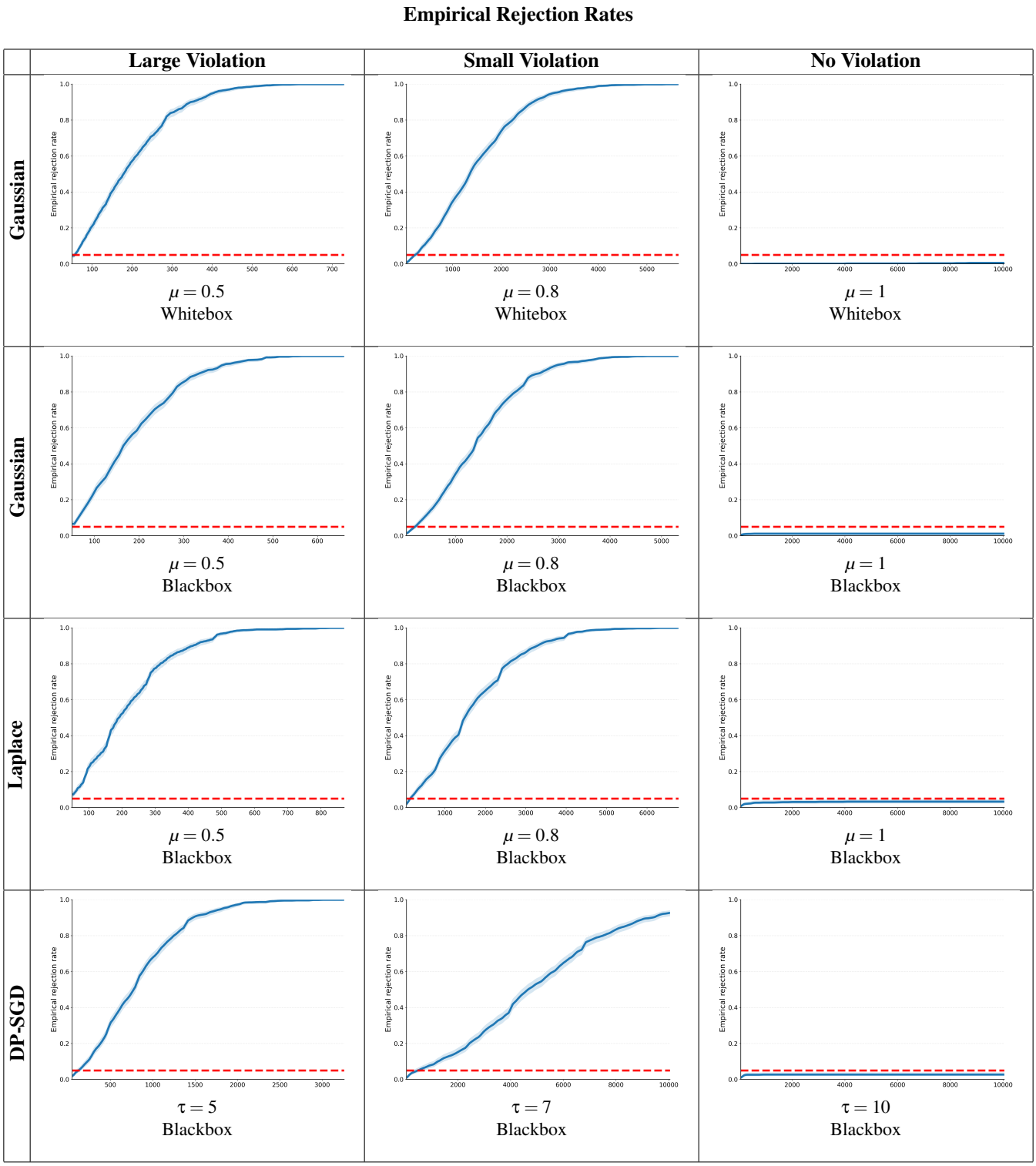


Figure 5: Empirical rejection rates relative to sample sizes across $1,000$ runs for a fixed significance level $\gamma = 0.05$. Rows correspond to different mechanisms and columns to the strength of privacy violation. Shaded ribbons indicate pointwise 95% confidence intervals.

MMD margin typically becomes small in larger-ε regimes, as the MMD threshold approaches $\tau_{\mathrm{MMD}}(\varepsilon,\delta)$. This "saturation effect" means that the effect size becomes small, which in turn reduces the test's power. Indeed, [22] report poor behavior beyond roughly $\varepsilon > 0.6$ in their DP-SGD experiments (so fairly high privacy regimes). Our auditor does not rely on this MMD-to-DP surrogate and therefore remains effective for more general privacy parameters. Conversely, MMD-based methods may be advantageous for certain high-dimensional settings with an appropriate kernel choice, whereas a KDE-based APT instantiation is more suited to low-dimensional settings. With these comments in mind, the below experiments should be interpreted as illustrating how the two sequential auditors behave under one specific test constructions, that was proposed by [22] to evaluate their method.

Additive-noise benchmark. For a direct quantitative comparison in the small-ε regimes where [22] report strong performance, we follow their additive-noise mean-estimation benchmarks. To align the sample-efficiency comparison with the privacy object audited by [22], we do not audit a tight tradeoff curve here as the preceding experiments . Instead, for the same claimed (ε,δ)-DP guarantee, our auditor audit the corresponding tradeoff lower bound

$$T_{\varepsilon,\delta}(\alpha) = \max\left\{0,\, 1-\delta-e^{\varepsilon}\alpha,\, e^{-\varepsilon}(1-\delta-\alpha)\right\}.$$

This choice is less tailored and may worsen our sample efficiency, but it makes the comparison with the (ε,δ)-DP audit of [22] more direct.
We now present the following randomized algorithms from [22]

$$\mathrm{DPLaplace}(X) := \frac{\sum_{i=1}^{n} X_i}{\tilde{n}} + \rho_1,$$
$$\mathrm{NonDPLaplace1}(X) := \frac{\sum_{i=1}^{n} X_i}{n} + \rho_2.$$

Here, $\tilde{n} = \max\{10^{-12}, n+\tau\}$ with $\tau \sim \mathrm{Laplace}(0, 2/\varepsilon)$, $\rho_1 \sim \mathrm{Laplace}(0, 2/\lceil \tilde{n}\varepsilon \rceil)$, and $\rho_2 \sim \mathrm{Laplace}(0, 2/\lceil n\varepsilon \rceil)$. Analogously, [22] define DPGaussian and NonDPGaussian, which should not be confused with the standard equal-variance Gaussian shift mechanism associated with μ-GDP. In particular, we use the same claimed $T_{\varepsilon,\delta}$ described above but not the Gaussian/GDP curve. We notice in passing that the adjacency notion in [22] deviates slightly from ours (exchanging records from a database vs. removing one), but this has little impact on the auditing itself. Below, we consider the remove database notion, in the choice of databases $D = (0), D' = (0,1)$ by [22]. We summarize the results of our simulation across $1,000$ runs by reporting the average rejection time $\bar{n}$ and the empirical standard deviation of these runs. Results are reported in Table 1. For ease of comparison, we also cite the corresponding numbers by [22] in the same table.

Interpretation of Table 1. In these small-ε regimes, both procedures detect the non-private Gaussian and Laplace benchmarks with rejection rate one. The stopping times differ substantially: for $\varepsilon = 0.01$, our average stopping times are approximately 23% and 25% of the paper-reported values in [22] for NonDPGaussian1 and NonDPLaplace1, respectively. For $\varepsilon = 0.1$, the corresponding ratios are approximately 13% and 13%. This supports the empirical claim that, on these additive-noise benchmarks and in the regimes reported in [22], our f-DP auditor can reject with fewer samples.
The DP-compliant controls also behave as expected. The (theoretical) significance level for both procedures is $\gamma = 0.05$. In finite samples, the observed rejection rates fluctuate due to randomness across simulations and for our $1,000$ runs, the binomial sampling variability suggests that values below 0.064 should be considered acceptable. Rejection rates by both methods are always well below 0.064. For [22], the reported rejection rates are exactly zero under H_0 in all scenarios. This is consistent with a conservative test, which can be desirable for false-positive control but is also associated with lower power or slower detection under alternatives.

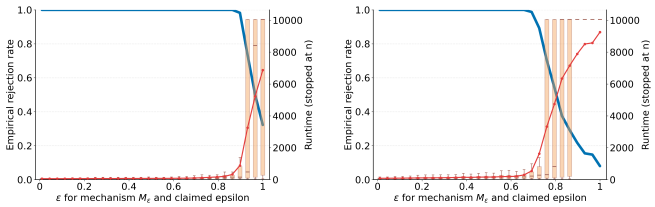


Figure 6: Larger-ε behavior of Algorithm 1 on NonDPGaussian1 (left) and NonDPLaplace1 (right). Blue curves show empirical rejection rates; red curves and boxplots show stopping times, capped at 10,000 samples.

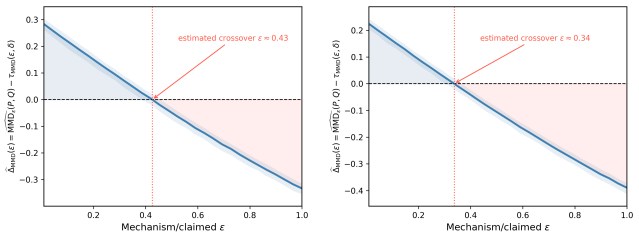


Figure 7: Estimated MMD margins for NonDPGaussian1 (left) and NonDPLaplace1 (right); shaded ribbons denote 95% confidence intervals. Once the margin crosses zero, around $\varepsilon \approx 0.43$ and $\varepsilon \approx 0.34$, respectively, the MMD-based approach can no longer reliably detect the violations.

Larger-ε regimes and the MMD saturation effect. We also sweep over larger claimed privacy levels for the two non-private mechanisms, NonDPGaussian1 and NonDPLaplace1,

Table 1: Sample efficiency comparison for the additive-noise benchmark. The left block reports Algorithm 1 over 1,000 repetitions, using the tradeoff lower bound induced by $(\varepsilon, 10^{-5})$-DP for Gaussian and by pure ε-DP for Laplace. The right block contains the paper-reported values from Table 1 of [22].

Mechanism	Algorithm 1 (ours)				Paper-reported values from [22]			
	$\varepsilon = 0.01$		$\varepsilon = 0.1$		$\varepsilon = 0.01$		$\varepsilon = 0.1$	
	Rej. rate	$\bar{n}$ to rej.	Rej. rate	$\bar{n}$ to rej.	Rej. rate	$\bar{n}$ to rej.	Rej. rate	$\bar{n}$ to rej.
DPGaussian	0	—	0	—	0	—	0	—
NonDPGaussian1	1	60.86 ± 19.33	1	71.60 ± 29.79	1	264 ± 9.3	1	562 ± 29.2
DPLaplace	0	—	0.021	—	0	—	0	—
NonDPLaplace1	1	81.38 ± 39.43	1	115.34 ± 70.92	1	331 ± 14.5	1	920 ± 61.6

to explain the difference between the two auditors. Figure 6 shows that our auditor continues to detect violations over a wider range of ε than the range in which the MMD surrogate has a positive margin. The blue curve reports the empirical rejection rate, while the red curve and boxplots summarize the stopping time of the auditors, with monitoring capped at 10,000 samples. As ε increases, the effective violation margin becomes smaller. Consequently, although these mechanisms remain buggy, their violations of the audited lower-bound tradeoff become harder to detect under our auditor instantiation.

Figure 7 reports the corresponding estimated MMD margins $\widehat{\Delta}_{\mathrm{MMD}}(\varepsilon) = \widehat{\mathrm{MMD}}(P,Q) - \tau_{\mathrm{MMD}}(\varepsilon,\delta)$ for the same benchmark, with $\widehat{\mathrm{MMD}}(P,Q)$ computed using the estimation procedure of [22]. $\widehat{\Delta}_{\mathrm{MMD}}(\varepsilon)$ crosses zero at approximately $\varepsilon \approx 0.43$ for NonDPGaussian1 and $\varepsilon \approx 0.34$ for NonDPLaplace1. Once the MMD margins are non-positive, collecting more samples cannot make the MMD surrogate reject reliably even if the mechanism still violates (ε,δ)-DP.

4.4 Real-World DP-SGD / Auditing in one run

In this section, we demonstrate that our sequential pipeline fits together with the recent emerging one-run DP-SGD auditing paradigm [38], in which a single DP-SGD training [1, 15] run simultaneously evaluates many (Dirac) canaries and provides multiple sequences of per-training-step observations for auditing. Concretely, we instantiate our method in a realistic DP-SGD setting using the now-standard whitebox canary methodology [37] from recent DP-SGD auditing work: the auditor injects Dirac canaries and records step-wise intermediate statistics from the noised and clipped gradient during the private training.

Following [37], at each DP-SGD step t, we extract a scalar observation for a fixed canary by taking the dot product between the (clipped and noised) gradient and a fixed canary direction. We then aggregate the per-step observations of a canary into a single scalar by averaging over training steps. We denote these averaged scores by $\{X_i\}_{i=1}^n$ for the canary-included world and by $\{Y_i\}_{i=1}^n$ for the no-canary world, where n is the number of (independent) canaries used per side. The Dirac canary direction is chosen to create a detectable gradient-inner-product signal for the resulting neighboring pair, which does not necessarily yield a worst-case neighboring pair.

To instantiate our sequential pipeline in Algorithm 1, it suffices to define a classifier using BUILDCLASSIFIER from samples $\{X_i\}$ and $\{Y_i\}$. We adopt the simple threshold rules presented in Section 4.2, where $\{\phi_\eta\}_{\eta\in\mathbb{R}}$ is given by

$$\phi_\eta(z) = \mathbf{1}\{z \geq \eta\}.$$

We have trained a DP model on the CIFAR-10 dataset following the architecture and hyperparameters of [15]: WideResNet-16-4 with GroupNorm/Weight-Standardization, batch size 4096, clipping norm $C = 1.0$, noise multiplier $\sigma = 3.0$. We additionally insert 10,000 canaries during training, following the canary setup of [37] [3]. As in the one-run auditing methodology, the Dirac canaries are placed on distinct coordinates, which reduces cross-canary interference and makes the resulting scores closer to the independent-sample setting considered in our theory. We nevertheless do not claim robustness to arbitrary dependence among canary scores.

Experiment settings. To illustrate the impact of sequential auditing, we run the following experiment. Denote by $\widehat{P}$ and $\widehat{Q}$ the empirical distributions of the canary-included and no-canary scores produced by the single training run. We then perform 250 independent resampling repetitions, and run our sequential auditor on each resampled subsets, respectively. We use these repetitions to study the auditor's variability without retraining DP-SGD, enabling the boxplots in Figure 8.

For the parameters, we choose the default settings as throughout the whole section. In Figure 8 we illustrate the

[3] Prior DP-SGD auditing work often uses $\approx 5{,}000$ canaries. We observe comparable behavior at 5,000; we use up to 10,000 here to more clearly illustrate the gains from sequential auditing over a wider range of claimed μ and number of canaries.

empirical rejection rates (blue) and average runtime until rejection, expressed by the sample size n (red). The x-axis represents the claimed μ (smaller μ means a larger DP violation). Additionally, we include boxplots of the runtimes over the 250 runs, to illustrate variation in runtimes. In this setup, the precise DP level is unknown. Yet a theoretical upper bound, in terms of (ε, δ)-DP, can be given with $\varepsilon^* = 6.56, \delta^* = 10^{-5}$.[4] Using the relation between $\mu-$GDP and $(\varepsilon, \delta)-$DP (see [17]), this bound translates into a parameter value of exactly $\mu = 1.4$; this means we expect that for $\mu = 1.4$, no DP violation exists. By the same calculation, we notice that $\mu = 1.2$ corresponds to $\varepsilon = 5.413$ and $\delta = 10^{-5}$, and $\mu = 1.1$ corresponds to $\varepsilon = 4.88$ and $\delta = 10^{-5}$.

Interpretation of the results First, consider the blue curve of rejection probabilities in Figure 8. At $\mu = 1.4$ (where we believe no DP violation exists), we do not reject in any of our 250 runs, and in particular hold the significance level. Afterwards, the number of rejections rises rapidly to about 35% for $\mu = 1.2$, reaching $> 99.5\%$ for $\mu = 1.1$. For smaller values of μ, i.e. larger DP violations, the rejection rate is always 100%, but larger violations are associated with shorter runtimes. To see this, consider the red curve of average runtimes until rejection. For $\mu = 1.4$, runtimes are maximal, because no rejection takes place (this is good because any rejection would be a false rejection in this case). Afterwards, average runtimes drop fast, to less than $3,000$ for $\mu = 1.1$ and going all the way to a few hundred for $\mu = 0.8$. *This result is crucial: it demonstrates that our adaptive DP auditor drastically reduces the sample size n, if small samples suffice.*

An even more nuanced picture emerges when considering the runtime boxplots, which express the variability in runtimes. Variability decreases as the DP violation grows larger, with all runs very short for $\mu \leq 1$. At $\mu = 1.1$ variability is higher, but we can see that about 50% of runs finish for $n \leq 2,000$, and 25% even for $n \leq 1,000$. Thus, even for smaller DP violations, there exists a large proportion of reasonably short audits. Boxplots for the cases $\mu \geq 1.2$ are less meaningful, because the majority of runs end without rejection.

Acknowledgments

Tim Kutta's work has been supported by the AUFF Nova grant 47222. Yu Wei and Vassilis Zikas were supported in part by NSF Award No. 2448339 and No. 2531010, Halcyon Futures via the AI Security Institute, JPMorgan Chase, the Stellar Development Foundation, and Sunday Group, Inc. Martin Dunsche was partially Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972

[4] This $\varepsilon^* = 6.56$ corresponding to $\delta^* = 10^{-5}$ is computed using the privacy accountant provided by `Opacus` [35] for DP-SGD, with the noise multiplier, sampling rate, and number of training steps derived from the prescribed training parameters.

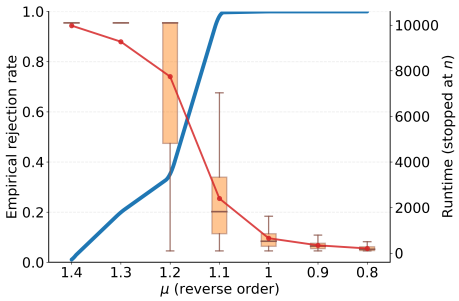


Figure 8: Empirical rejection rate (blue curve) and average runtime (red curve) for different claimed μ. Variation in runtimes is represented by orange boxplots. Notice that smaller values for μ correspond to larger privacy violation.

Ethical Considerations

Stakeholder analysis. The primary stakeholders are data subjects, whose privacy the audited mechanisms are designed to protect, model trainers seeking to validate their privacy guarantees. The intended benefit of this work is to reduce the cost of detecting implementation bugs, or invalid audit practices that could otherwise lead to excess privacy leakage.

Risks and mitigations. Likeur other privacy auditing tools, o auditor could in principle be misused to search for weak or misconfigured deployments. This risk is mitigated by the fact that the auditor requires repeated blackbox or whitebox access to the mechanism and reports only statistical evidence of a privacy violation, rather than reconstructing private records or revealing the underlying dataset.

Decision to conduct and publish the research. We conducted this work because sample-efficient and statistically valid privacy auditing can help practitioners identify and fix privacy failures before deployment. We publish the method and artifacts to support reproducibility and help the community avoid invalid early-stopping practices in DP auditing. Our intended use case is authorized auditing by mechanism designers, system owners, or compliance teams.

Open Science

To comply with the Open Science Policy, we make our research artifacts available to the public [5]. The artifact includes

[5] https://github.com/martindunsche/sequential_fdp_auditing; permanent archive: https://doi.org/10.5281/zenodo.20398258

the code implementing our auditing pipeline, together with scripts and instructions to reproduce the experimental results reported in the paper. In our repository, we ensure the transparency and reproducibility by describing the dataset used, the specification of our machines, and citing all algorithms tested.

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[2] John M. Abowd. The U.S. census bureau adopts Differential Privacy. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 2867, 2018.

[3] John M. Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson L. Garfinkel, Matthew Heineck, Christine Heiss, Robert Johns, Daniel Kifer, Patrick Leclerc, Ashwin Machanavajjhala, Benjamin Moran, William Sexton, Matthew Spence, and Pavel Zhuravlev. The 2020 Census Disclosure Avoidance System TopDown Algorithm. *Harvard Data Science Review*, 2022.

[4] Apple Differential Privacy Team. Learning with privacy at scale. Apple Machine Learning Research, 2017. https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf.

[5] Önder Askin, Holger Dette, Martin Dunsche, Tim Kutta, Yun Lu, Yu Wei, and Vassilis Zikas. General-purpose f-DP estimation and auditing in a black-box setting. In *Proceedings of the USENIX Security Symposium*, pages 2713–2732, 2025.

[6] Jordan Awan and Aishwarya Ramasethu. Optimizing noise for f-Differential Privacy via anti-concentration and stochastic dominance. *Journal of Machine Learning Research*, 25:1–32, 2024.

[7] Raef Bassily, Adam Groce, Jonathan Katz, and Adam D. Smith. Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 439–448, 2013.

[8] István Berkes, Weidong Liu, and Wei Biao Wu. Komlós–Major–Tusnády approximation under dependence. *The Annals of Probability*, 42(2):794–817, 2014.

[9] Benjamin Bichsel, Timon Gehr, Dana Drachsler-Cohen, Petar Tsankov, and Martin T. Vechev. DP-finder: Finding Differential Privacy violations by sampling and optimization. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 508–524, 2018.

[10] Benjamin Bichsel, Samuel Steffen, Ilija Bogunovic, and Martin T. Vechev. DP-sniper: Black-box discovery of Differential Privacy violations using classifiers. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 391–409, 2021.

[11] R. C. Bradley. *Introduction to Strong Mixing Conditions*, volume 1,2,3. Kendrick Press, 2007.

[12] Mark Bun and Thomas Steinke. Concentrated Differential Privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the Theory of Cryptography Conference*, volume 9985 of *Lecture Notes in Computer Science*, pages 635–658, 2016.

[13] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 1897–1914, 2022.

[14] Rachel Cummings, Damien Desfontaines, David Evans, Roxana Geambasu, Yangsibo Huang, Matthew Jagielski, Peter Kairouz, Gautam Kamath, Sewoong Oh, Olga Ohrimenko, Nicolas Papernot, Ryan Rogers, Milan Shen, Shuang Song, Weijie Su, Andreas Terzis, Abhradeep Thakurta, Sergei Vassilvitskii, Yu-Xiang Wang, Li Xiong, Sergey Yekhanin, Da Yu, Huanyu Zhang, and Wanrong Zhang. Advancing differential privacy: Where we are now and future directions for real-world deployment. *Harvard Data Science Review*, 6(1), 2024.

[15] Soham De, Leonard Berrada, Jamie Hayes, Samuel L. Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv:2204.13650*, 2022. Presented at TPDP Workshop at ICML 2022.

[16] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems*, pages 3574–3583, 2017.

[17] Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian Differential Privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022.

[18] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology — EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[19] Cynthia Dwork and Aaron Roth. The algorithmic foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[20] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 51–60. IEEE, 2010.

[21] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 1054–1067, 2014.

[22] Tomás González, Mateo Dulce Rubio, Aaditya Ramdas, and Mónica Ribero. Sequentially auditing Differential Privacy. In *Advances in Neural Information Processing Systems*, 2025.

[23] Moritz Hardt and Kunal Talwar. On the geometry of Differential Privacy. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, pages 705–714. ACM, 2010.

[24] Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *Annals of Statistics*, 49(2):1055–1080, 2021.

[25] Gautam Kamath and Jonathan Ullman. A Primer on Private Statistics. *arXiv:2005.00010*, 2020.

[26] János Komlós, Péter Major, and Gábor Tusnády. An approximation of partial sums of independent R.V.'s and the sample DF.I. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 32:111–131, 1975.

[27] János Komlós, Péter Major, and Gábor Tusnády. An approximation of partial sums of independent R.V.'s and the sample DF.II. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 34:33–58, 1976.

[28] T. Kutta and P. Kokoszka. Prokhorov metric convergence of the partial sum process for reconstructed functional data. *arXiv preprint arXiv:2506.21172*, 2025.

[29] Tim Kutta, Önder Askin, and Martin Dunsche. Lower bounds for Rényi Differential Privacy in a black-box setting. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 951–971, 2024.

[30] Erich L. Lehmann and Joseph P. Romano. *Testing Statistical Hypotheses*. Springer Texts in Statistics. Springer Science+Business Media, New York, NY, USA, 3rd edition, 2005.

[31] Xiyang Liu and Sewoong Oh. Minimax optimal estimation of approximate Differential Privacy on neighboring databases. In *Advances in Neural Information Processing Systems*, pages 2414–2425, 2019.

[32] Yun Lu, Malik Magdon-Ismail, Yu Wei, and Vassilis Zikas. Eureka: A general framework for black-box Differential Privacy estimators. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 913–931, 2024.

[33] Saeed Mahloujifar, Luca Melis, and Kamalika Chaudhuri. Auditing f-Differential Privacy in one run. *arXiv:2410.22235*, 2024.

[34] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 94–103. IEEE, 2007.

[35] Meta AI. Opacus: User-Friendly differential privacy library in PyTorch. https://opacus.ai/, 2026. Accessed 2026-02-05.

[36] Ilya Mironov. Rényi Differential Privacy. In *Proceedings of the IEEE Computer Security Foundations Symposium*, pages 263–275, 2017.

[37] 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 *Proceedings of the USENIX Security Symposium*, pages 1631–1648, 2023.

[38] Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy auditing with one (1) training run. *Advances in Neural Information Processing Systems*, 36:49268–49280, 2023.

[39] Salil P. Vadhan. The complexity of differential privacy. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer International Publishing, 2017.

[40] Matt Wand. *KernSmooth: Functions for Kernel Smoothing Supporting Wand & Jones (1995)*, 2025. R package version 2.23-26. Available at https://CRAN.R-project.org/package=KernSmooth.

[41] Chendi Wang, Buxin Su, Jiayuan Ye, Reza Shokri, and Weijie J. Su. Unified enhancement of privacy bounds for mixture mechanisms via f-Differential Privacy. In *Advances in Neural Information Processing Systems*, 2023.

[42] Yuxin Wang, Zeyu Ding, Daniel Kifer, and Danfeng Zhang. Checkdp: An automated and integrated approach for proving Differential Privacy or finding precise counterexamples. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 919–938, 2020.

[43] Hanshen Xiao and Srinivas Devadas. PAC privacy: Automatic privacy measurement and control of data processing. In *Advances in Cryptology — CRYPTO 2023*, volume 14082 of *Lecture Notes in Computer Science*, pages 611–644. Springer, 2023.

A Proofs and Technical Details

The main probabilistic result, used in our theory are the KMT [26,27] approximations. A modern formulation and extension of these results is given e.g. in [8]. The last reference matters in connection with our result to settings with dependent data, which we discuss in Appendix C. For the next result, we restrict our attention to i.i.d. data. As a first step we cite the KMT approximations, stated in the above papers.

Theorem A.1. (KMT approximation) Suppose $(X_i)_i$ are i.i.d. random variables with $|X_i| \le C < \infty$ a.s., $\mathbb{E}X_i = 0$ and $\mathbb{E}X_i^2 =: \sigma^2$. Then, possibly after enlarging the probability space, there exist i.i.d. Gaussians $(Z_i)_i$ with $Z_i \sim \mathcal{N}(0,\sigma^2)$ such that

$$\left|\sum_{i=1}^{k}\{X_i - Z_i\}\right| = O(\log(k)), \qquad a.s.$$

We use the KMT approximation to conduct, roughly speaking, an infinite number of hypothesis tests in k.

A.1 Proof of Theorem 3.1

The main step in the proof will be the Gaussian approximation of the difference of empirical errors of ϕ minus the true error. We demonstrate the main step for the error α_ϕ, noting that the proof for β_ϕ works in exact analogy. For any $k \ge M$, the estimator of α_ϕ is the empirical version $\hat{\alpha}_\phi(k) := \frac{1}{k}\sum_{i=1}^{k}\phi(X_i)$. We now consider the rescaled difference

$$\begin{aligned} D(k) :=& \frac{\sqrt{k}}{\sqrt{\log(20+k/M)}}\left[\hat{\alpha}_\phi(k) - \alpha_\phi\right] \\ =& \frac{1}{\sqrt{k\log(20+k/M)}}\sum_{i=1}^{k}\left[\phi(X_i) - \mathbb{E}\phi(X_i)\right]. \end{aligned}$$

Since $\phi(x) \in [0,1]$, the random variables $\phi(X_i) - \mathbb{E}\phi(X_i)$ are independent, identically distributed and bounded. In particular, using Theorem A.1, we can find normally distributed random variables $Z_i \sim \mathcal{N}(0, \mathbb{V}ar(\phi(X_1)))$ for $1 \le i < \infty$ such that almost surely

$$\left|\sum_{i=1}^{k}\left\{\left[\phi(X_i) - \mathbb{E}\phi(X_i)\right] - Z_i\right\}\right| = O(\log(k))$$

Defining $D_Z(k) := \frac{1}{\sqrt{k\log(20+k/M)}}\sum_{i=1}^{k} Z_i$, it follows that, for large M

$$\begin{aligned} \sup_{k\ge M}|D_Z(k) - D(k)| &= O\left(\sup_{k\ge M}\frac{\log(k)}{\sqrt{k\log(20+k/M)}}\right) \\ &= O\left(\sup_{k\ge M}\frac{\log(k)/k^{1/4}}{M^{1/4}}\right) = O(M^{-1/4}) = o(1) \end{aligned}$$

almost surely. On an intuitive level, this approximation completes the proof. We first explain why and then provide some remaining mathematical details. We have shown that $D(k)$ may be replaced for all $k \ge M$ by $D_Z(k)$. Now, taking the upper $(1-\gamma/2)$ quantile of $\sup_{k\ge M} D_Z(k)$, say $q_{1-\gamma/2}$, implies that

$$D(k) \le q_{1-\gamma/2}$$

for all $k \ge M$ simultaneously with probability converging to $1-\gamma/2$ as $M \to \infty$. The quantile is calculated by Algorithm 4. In practice there exists a minor gap, since $\mathbb{V}ar(\phi(X_1))$ is not known, but it can be estimated by the empirical variance over the burn-in period and Slutsky's theorem ensures consistency. Using an analogous approximation for β_ϕ, we obtain a sequence of one-sided confidence rectangles $(\alpha_\phi, \beta_\phi)$ described by Algorithm 4. Coverage is, by the Bonferroni bound, $1-\gamma$ for $M \to \infty$, and the confidence bounds converge to $(\alpha_\phi, \beta_\phi)$ as $k \to \infty$. Accordingly, the guarantees from Theorem 3.1 follow directly. One mathematical subtlety remains: For our above arguments, we have used the real-valued quantiles of the object $\sup_{k\ge M} D_Z(k)$. Yet we have, strictly speaking, not established that these are finite (or uniformly bounded in M). To ensure that, we show weak convergence of $\sup_{k\ge M} D_Z(k)$, which in particular implies finite quantiles. A bit of mathematical background is therefore needed on the so-called Brownian motion, a key process in probability theory. The Brownian motion is here denoted by $W(t), 0 \le t < \infty$. It is a random function and we list some of its properties, useful for our below derivation:

B1) $W(0) = 0$ a.s. and $t \mapsto W(t)$ is continuous a.s.

B2) W has independent, centered and Gaussian distributed increments.

B3) $\mathrm{Cov}(W(s), W(t)) = \min\{s,t\} \cdot \mathbb{V}ar(W(1))$.

B4) $W(t), t \ge 0$ has the same distribution as $W(ct)/\sqrt{c}, t \ge 0$ for any $c > 0$.

B5)
$$\limsup_{t\to\infty}\frac{|W(t)|}{\sqrt{2t\log\log t}} = \sqrt{\mathbb{V}ar(W(1))}, \quad a.s.$$

Properties B1)-B3) are commonly used to characterize the Brownian motion. B4) and B5) are important consequences. B4) is often referred to as the scaling property, and intuitively states that the Brownian motion looks the same on large or small scales. B5) is called "the law of the iterated logarithm"

and ensures that $W(t)$ is (eventually) contained in a domain only growing slightly faster than $\sqrt{t}$. Returning to our proof, we notice that the following equality holds in distribution

$$\sup_{k\geq M} D_Z(k) \stackrel{d}{=} \sup_{k\geq M} \frac{W(k)}{\sqrt{k\log(20+k/M)}}.$$

Here, W is a centered Brownian motion, with variance $\sigma^2 := \mathbb{V}ar(\phi(X_1))$. The reason is that (using B1)-B3)) one can easily show that $\sum_{i=1}^{k} Z_i, k \geq 1$ have jointly the same distribution as $W(k), k \geq 1$. Next, using the rescaling property B4) with $c = 1/M$ it follows that in distribution

$$\sup_{k\geq M} \frac{W(k)}{\sqrt{k\log(20+k/M)}} \stackrel{d}{=} \sup_{x\geq 1, x=k/M} \frac{W(x)}{\sqrt{x\log(20+x)}}.$$

The random variable on the right is almost surely finite, due to the law of iterated logarithm for the Brownian motion in B5). Now, let us fix the random seed and consider $W(t), t \geq 0$ just as a fixed function. As $M \to \infty$, the right side converges (monotonically from below) to $L := \sup_{x\geq 1} \frac{W(x)}{\sqrt{x\log(20+x)}}$. To obtain this convergence, we have used that the function $W(x)/(\sqrt{x\log(20+x)}$ is continuous according to B1) and goes to 0 as $x \to \infty$ which is again following from an application of the law of the iterated logarithm B5). These considerations imply the weak convergence $\sup_{k\geq M} D_Z(k) \stackrel{d}{\to} L$, finishing our proof.

B Additional Algorithms

Our APT procedure makes use of a variety of subprocedures, e.g. to calculate confidence boundaries or to build classifiers in various scenarios. We give a list of blueprints below.

Algorithm 2 BUILDCLASSIFIER: Generic classifier construction for APT via a learned score function.

Require: Samples $\{X_i\}_{i=1}^m$ and $\{Y_i\}_{i=1}^m$; candidate threshold set $\mathcal{T}$; claimed curve f; learning rule $\mathcal{L}$.

Ensure: A classifier $\phi(\cdot)$ and its selected threshold η^*.

function BUILDCLASSIFIER($\{X_i\}_{i=1}^m, \{Y_i\}_{i=1}^m, \mathcal{T}, f, \mathcal{L}$)
  Train a scoring function $s(\cdot) \leftarrow \mathcal{L}(\{X_i\}_{i=1}^m, \{Y_i\}_{i=1}^m)$.
  for each $\eta \in \mathcal{T}$ **do**
    Define $\phi_\eta(y) \leftarrow \mathbf{1}\{s(y) \geq \eta\}$.
    Compute $\hat{\alpha}(\eta) \leftarrow \frac{1}{m}\sum_{i=1}^m \phi_\eta(X_i)$.
    Compute $\hat{\beta}(\eta) \leftarrow 1 - \frac{1}{m}\sum_{i=1}^m \phi_\eta(Y_i)$.
  end for
  Select η^* using equation (8) applied to the curve $\{(\hat{\alpha}(\eta), \beta(\eta)) : \eta \in \mathcal{T}\}$.
  Define $\phi(y) \leftarrow \mathbf{1}\{s(y) \geq \eta^*\}$.
  return $\phi(\cdot)$.
end function

Algorithm 3 PLRT: Non-parametric estimator for an optimal f-DP classifier

Require: Blackbox access to $\mathcal{M}$; threshold $\eta > 0$; sample size n; databases D, D'.

Ensure: Estimate $(\hat{\alpha}(\eta), \hat{\beta}(\eta))$ of $(\alpha(\eta), \beta(\eta))$ for (P, Q), where $\mathcal{M}(D) \sim P$ and $\mathcal{M}(D') \sim Q$.

Choose perturbation parameter h.
Set density estimator $\mathcal{A}$ (default: KDE).
function PLRT($\mathcal{A}, \mathcal{M}, D, D', \eta_{\text{vec}}, n$)
  Run $\mathcal{M}$ on D and D' to obtain n samples from each.
  Fit densities $\hat{p}$ and $\hat{q}$ using $\mathcal{A}$.
  for $\eta \in \eta_{\text{vec}}$ **do**
    $\hat{\alpha}(\eta) \leftarrow \frac{1}{h}\int_{-h/2}^{h/2} \left(\int \mathbf{1}\left\{ \log\frac{\hat{q}(z)}{\hat{p}(z)} > \log\eta + x \right\} \hat{p}(z)\, dz \right) dx$
    $\hat{\beta}(\eta) \leftarrow 1 - \frac{1}{h}\int_{-h/2}^{h/2} \left(\int \mathbf{1}\left\{ \log\frac{\hat{q}(z)}{\hat{p}(z)} > \log\eta + x \right\} \hat{q}(z)\, dz \right) dx$
  end for
  return $(\hat{p}, \hat{q}, \hat{T})$ where $\hat{T} = \{(\hat{\alpha}(\eta), \hat{\beta}(\eta)) : \eta \in \eta_{\text{vec}}\}$.
end function

Algorithm 4 Confidence Adjustment for Sequential Audit

Require: Empirical means $\hat{T}_1, \hat{T}_2$; critical value $q_{1-\gamma/2}$; burn-in size M; current sample size k.

Ensure: Confidence-adjusted $(\hat{T}_\alpha, \hat{T}_\beta)$.

function CONFADJ($\hat{T}_1, \hat{T}_2, q_{1-\gamma/2}, M, k$)
  $b \leftarrow \frac{\sqrt{\log(20+k/M)}}{\sqrt{k}}$, standard deviations $s_1 \leftarrow \sqrt{\hat{T}_1(1-\hat{T}_1)}$, $s_2 \leftarrow \sqrt{\hat{T}_2(1-\hat{T}_2)}$.
  $\hat{T}_\alpha \leftarrow T_1 + q_{1-\gamma/2}\, s_1\, b$; $\hat{T}_\beta \leftarrow T_2 + q_{1-\gamma/2}\, s_2\, b$
  $\hat{T}_\alpha \leftarrow \min\{1, \max\{0, \hat{T}_\alpha\}\}$
  $\hat{T}_\beta \leftarrow \min\{1, \max\{0, \hat{T}_\beta\}\}$
  return $(\hat{T}_\alpha, \hat{T}_\beta)$
end function

Algorithm 5 QUANT(M, α), where standard values $\alpha = 0.01, 0.05, 0.1$ are saved.

Require: Burn-in size $M \in \mathbb{N}$, significance level $\alpha \in (0,1)$

Ensure: $(1-\alpha)$-quantile $q_{1-\alpha}$ of $W = \sup_{k\geq M} \frac{1}{\sqrt{(k)\log(20+k/M)}} \sum_{i=1}^{k} Z_i$

$R \leftarrow 10^5$ ▷ number of Monte-Carlo replications
$T \leftarrow 10^4$ ▷ truncation for the supremum
for $r = 1, \ldots, R$ **do**
  Draw $Z_1^{(r)}, \ldots, Z_T^{(r)}$ i.i.d. $\mathcal{N}(0,1)$
  $W_r \leftarrow \max_{k\leq M} \frac{1}{\sqrt{k\log(20+k/M)}} \sum_{i=1}^{k} Z_i^{(r)}$
end for
$q_{1-\alpha} \leftarrow$ empirical $(1-\alpha)$-quantile of $\{W_1, \ldots, W_R\}$
return $q_{1-\alpha}$

Algorithm 6 COMPUTE-η^*-DIAGONAL: 45°-based selection of critical η^*

Require: Estimated tradeoff points $(\hat{\alpha}(\eta), \hat{\beta}(\eta))$ for all $\eta \in \eta_{\text{vec}}$, and the claimed curve f.

0.1cm

Ensure: Critical η^* maximizing diagonal (45°) distance between empirical and claimed curves.

function COMPUTE-η^*-DIAGONAL($\hat{\alpha}(\eta), \hat{\beta}(\eta), f, \eta_{\text{vec}}$)
  Initialize dists $\leftarrow 0$.
  for η_i in η_{vec} **do**
    Set $\hat{\alpha} \leftarrow \hat{\alpha}(\eta_i)$ and $\hat{\beta} \leftarrow \hat{\beta}(\eta_i)$.
    Define $f_{45}(\alpha') := f(\alpha') - [\hat{\beta} + (\alpha' - \hat{\alpha})]$.
    Solve for α' satisfying $f_{45}(\alpha') = 0$ on $[0,1]$ ▷ (e.g., via `uniroot`).
    if a valid root α' is found **then**
      Set $\beta' \leftarrow f(\alpha')$.
      Compute $d(\eta_i) := \sqrt{(\alpha' - \hat{\alpha})^2 + (\beta' - \hat{\beta})^2}$.
    else
      Set $d(\eta_i) :=$ NA.
    end if
  end for
  Return $\eta^* := \arg\max_{\eta_i \in \eta_{\text{vec}}} d(\eta_i)$.
end function

Algorithm 7 AUDITSHOULDREFIT: Dynamic choice of refresh period M_1.

Require: Last refit sample size n_{last}; refit threshold $\tau \in (0,1)$.
Ensure: M_1.

function AUDITSHOULDREFIT(n_{last}, τ)
  Compute shrinkage for $n > n_{\text{last}}$:

$$\texttt{shrink}(n) \leftarrow 1 - \left(\frac{n_{\text{last}}}{n}\right)^{1/5}.$$

  Set decision: $M_1 \leftarrow \arg\min_n \mathbf{1}\{\texttt{shrink}(n) > \tau\}$.
  return M_1.
end function

C Theory for Adaptive Classifiers

In Theorem 3.1, we justified our sequential auditing procedure under the simplifying assumption that the classifier ϕ is fixed throughout the audit. This assumption is useful for explaining the core idea, but it is restrictive in blackbox settings. There, the analyst will typically refine the classifier as more samples become available. To justify such scenarios mathematically, we derive in this section a Gaussian approximation for the estimated type-I and type-II errors under evolving classifiers. This approximation is (as we recall) the central tool for proving Theorem 3.1.

The key point is the following. Reusing previous samples to update the classifier makes the sequence of classification outcomes dependent. However, sequential Gaussian approximations are not limited to independent data. They remain valid for weakly dependent and approximately stationary processes under suitable assumptions. We therefore formulate the adaptive classifier sequence as a weakly dependent process and invoke the Gaussian approximation principle developed in [28]. The resulting theorem is an adaptive analogue of Theorem 3.1.

Let $(X_k, Y_k)_{k\geq 1}$ denote the pairs of outputs sampled from the two neighboring databases, with $X_k \sim P$ and $Y_k \sim Q$. At time k, the classifier may depend on all previous observations, but not on the current pair (X_k, Y_k). We write

$$\hat{\phi}_k(x) = \mathcal{H}_k\big(x, \varepsilon_k, (X_{k-1}, Y_{k-1}), \ldots, (X_1, Y_1)\big),$$

where $\mathcal{H}_k$ is measurable and where $(\varepsilon_k)_{k\geq 1}$ is an iid sequence, independent of the data, representing possible randomization used by the classifier construction. The classification outcome at time k is encoded by

$$R_k := \begin{pmatrix} \hat{\phi}_k(X_k) \\ 1 - \hat{\phi}_k(Y_k) \end{pmatrix} \in \{0,1\}^2.$$

Thus the empirical error pair used by the auditor is $\hat{\theta}_k := \begin{pmatrix} \hat{\alpha}_k \\ \hat{\beta}_k \end{pmatrix} := \frac{1}{k}\sum_{i=1}^{k} R_i$. The first component estimates the type-I classification error and the second component estimates the type-II classification error. We next state the assumptions under which adaptive monitoring remains valid. For this purpose, we first need a notion of dependence across data.

Definition C.1. Let $(Z_k)_{k\geq 1}$ be a time series with values in a separable metric space. Its α-mixing coefficients are defined by

$$\alpha_Z(h) := \sup_{m\geq 1} \sup_{\substack{A\in\sigma(Z_1,\ldots,Z_m) \\ B\in\sigma(Z_{m+h},Z_{m+h+1},\ldots)}} |\mathbb{P}(A\cap B) - \mathbb{P}(A)\mathbb{P}(B)|.$$

We say that $(Z_k)_{k\geq 1}$ is α-mixing if $\alpha_Z(h) \to 0$ as $h \to \infty$.

The intuition of mixing coefficients is simple. In the case where the two σ-algebras are independent, it should hold for any pair of events that $\mathbb{P}(A\cap B) = \mathbb{P}(A)\mathbb{P}(B)$, or equivalently that $\mathbb{P}(A\cap B) - \mathbb{P}(A)\mathbb{P}(B) = 0$. Consequently, a natural generalization to weak dependence, where σ-algebras contain little information about each other, is that the difference be small. A detailed discussion of mixing can be found in [11]. Mixing has for a long time been the most popular way to describe dependence. Nowadays there are alternatives, such as physical dependence, and we conjecture that similar results as derived below would also hold under this dependence concept. Next, we formulate the main technical assumptions for this section.

Assumption C.2. The adaptive classification process $(R_k)_{k\geq 1}$ satisfies the following conditions.

A1) The process $(R_k)_{k\geq 1}$ is α-mixing with coefficients satisfying

$$\alpha_R(h) \leq C_\alpha h^{-p}$$

for some constants $C_\alpha > 0$ and $p > 4$.

A2) There exists a deterministic measurable classifier ϕ such that, with $R_k^\circ := \begin{pmatrix} \phi(X_k) \\ 1-\phi(Y_k) \end{pmatrix}$, we have

$$\mathbb{P}(R_k \neq R_k^\circ) \leq C_\phi k^{-\nu}$$

for some constants $C_\phi, \nu > 0$.

Assumption A1 means that the dependence induced by reusing earlier observations for classifier updates is weak enough. The assumption of polynomial decay of mixing coefficients is relatively weak. Assumption A2 means that the adaptive classifier stabilizes asymptotically toward a deterministic classifier ϕ. We do not assume that ϕ is optimal. Optimality is relevant for power, but not for validity of the sequential confidence adjustment.

We next prove the main Gaussian approximation needed to justify the adaptive auditor. The statement is scalar and applies to either coordinate of R_k. This is sufficient, since the confidence adjustment in the auditor is applied coordinatewise.

For $j \in \{1,2\}$, let $e_1 = (1,0)^\top$ and $e_2 = (0,1)^\top$, and define

$$U_{j,k} := e_j^\top R_k, \qquad U_{j,k}^\circ := e_j^\top R_k^\circ.$$

Thus

$$U_{1,k} = \hat{\phi}_k(X_k), \qquad U_{2,k} = 1 - \hat{\phi}_k(Y_k).$$

Write $S_j(k) := \sum_{i=1}^k \big(U_{j,i} - \mathbb{E}U_{j,i}\big)$ and

$$D_{j,M}(k) := \frac{S_j(k)}{\sqrt{k \log(20 + k/M)}}, \qquad k \geq M.$$

Lemma C.3 (Proof is in the full version)**.** Suppose Assumption C.2 holds. Fix $j \in \{1,2\}$ and assume that

$$\sigma_j^2 := \mathrm{Var}(U_{j,1}^\circ) > 0.$$

Then, on a suitable probability space,

$$\sup_{k \geq M} D_{j,M}(k) = \sup_{x \geq 1} \frac{\sigma_j W_j(x)}{\sqrt{x \log(20+x)}} + o_P(1), \qquad M \to \infty,$$

where W_j is a standard Brownian motion.

We now prove the bounded law of the iterated logarithm that was used in the previous proof. The next result uses the definition of the Prokhorov metric, taken from [28].

Definition C.4. Let $(\mathcal{M}, d_{\mathcal{M}})$ be a generic metric space. We define for a set $A \subset \mathcal{M}$ and $\chi > 0$ the χ-environment

$$A^\chi := \{x \in \mathcal{M} \mid \exists a \in A : d_{\mathcal{M}}(x,a) < \chi\}.$$

We call a set A measurable if it is a Borel set in $\mathcal{M}$ with respect to $d_{\mathcal{M}}$. Now we can define the Prokhorov distance of two probability measures $\mathbb{P}_1, \mathbb{P}_2$ as

$$\pi(\mathbb{P}_1, \mathbb{P}_2) :=$$
$$\inf\Big\{\chi > 0 : \mathbb{P}_1(A) \leq \mathbb{P}_2(A^\chi) + \chi \text{ and } \mathbb{P}_2(A) \leq \mathbb{P}_1(A^\chi) + \chi \ \forall A\Big\}.$$

The Prokhorov metric is a somewhat technical concept: it measures the distance between two probability measures, taking into account the topology of the underlying metric space. This is different from alternatives such as the total variation metric, which poses stricter requirements that do not, in the same way, refer to topology. A drawback of total variation is that total variation convergence often fails to hold, even when weak convergence holds. In contrast, Prokhorov convergence is closely related to weak convergence; on separable spaces it is even equivalent to it. It therefore gives a natural metric for proximity of distributions.

In the next lemma, we impose proximity of two stochastic processes with respect to the Prokhorov metric. The underlying metric space is that of bounded functions on the unit interval $[0,1]$, equipped with the supremum norm. We also use the notation that, with only minor abuse, we write $\pi(X,Y)$ for random variables X, Y when we mean $\pi(\mathbb{P}_X, \mathbb{P}_Y)$, identifying random variables with their distributions. The proof is an adaptation of the proof strategy of Corollary 2 in [28].

For the next lemma, we define for a general triangular array of bounded random variables $(U_{i,M})_{i,M}$ the sum

$$S'_M(k) = \sum_{i=1}^k \big(U_{i,M} - \mathbb{E}U_{i,M}\big).$$

Lemma C.5 (Proof is in the full version)**.** Define the partial sum process

$$P_{M,n}(t) := \frac{1}{\sqrt{n}} S'_M(\lfloor nt \rfloor), \qquad 0 \leq t \leq 1.$$

Suppose it satisfies, uniformly over M, the Prokhorov approximation bound

$$\pi\big(P_{M,n}, \sigma W\big) \leq C n^{-\tau} \tag{PA}$$

for some constants $C, \tau > 0$, where W is standard Brownian motion on $[0,1]$ and $\sigma^2 > 0$. Also assume that the individual variables $U_{i,M}$ are bounded. Then

$$\sup_{k \geq 1} \left| \frac{S'_M(k)}{\sqrt{k \log(\log(k+e))}} \right| = O_P(1), \qquad M \to \infty.$$

For the next result, suppose that Algorithm 1 is run with the outputs of the adaptive classifiers R_k described in this section. Recall that ϕ denotes the limiting classifier appearing in Assumption C.2.

Theorem C.6 (Proof is in the full version)**.** Suppose the assumptions of Lemma C.3 hold for both coordinates $j = 1,2$.

a) Suppose that the claimed f-DP guarantee is satisfied. Then

$$\limsup_{M \to \infty} \mathbb{P}(\text{APT rejects for some } k \geq M) \leq \gamma.$$

b) Suppose that the claimed f-DP guarantee is violated and that $f(\alpha_\phi) > \beta_\phi$. Then, for $k_{\max} = \infty$,

$$\lim_{M \to \infty} \mathbb{P}(\text{APT rejects for some } k \geq M) = 1.$$