Monitoring Violations of Differential Privacy over Time

Önder Askin
Ruhr University Bochum
oender.askin@rub.de

Tim Kutta
Aarhus University
tim.kutta@math.au.dk

Holger Dette
Ruhr University Bochum
holger.dette@rub.de

Abstract

Auditing differential privacy has emerged as an important area of research that supports the design of privacy-preserving mechanisms. Privacy audits help to obtain empirical estimates of the privacy parameter, to expose flawed implementations of algorithms and to compare practical with theoretical privacy guarantees. In this work, we investigate an unexplored facet of privacy auditing: the sustained auditing of a mechanism that can go through changes during its development or deployment. Monitoring the privacy of algorithms over time comes with specific challenges. Running state-of-the-art (static) auditors repeatedly requires excessive sampling efforts, while the reliability of such methods deteriorates over time without proper adjustments. To overcome these obstacles, we present a new monitoring procedure that extracts information from the entire deployment history of the algorithm. This allows us to reduce sampling efforts, while sustaining reliable outcomes of our auditor. We derive formal guarantees with regard to the soundness of our methods and evaluate their performance for important mechanisms from the literature. Our theoretical findings and experiments demonstrate the efficacy of our approach.

1 Introduction

Differential privacy (DP) [1] is a popular approach for bounding the information leakage associated with data-processing algorithms. If used appropriately, differentially private mechanisms can ensure the privacy of individuals that are represented in a dataset. DP algorithms are therefore adopted by companies and public agencies in settings where sensitive data is involved [2, 3, 4]. Yet, the development and deployment of DP mechanisms are susceptible to errors, which can result in algorithms that do not meet their purported level of privacy [5, 6, 7]. Auditing procedures can help to detect and remedy these privacy violations, which makes them useful tools for the design and implementation of DP algorithms.

Auditing methods vary with regard to their design and objectives. Some approaches, for instance, require access to the algorithm's code [8, 9, 10], while others limit themselves to the inputs and outputs of the mechanism under investigation [11, 12, 13]. A number of tools are designed for variants of differential privacy (rather than pure DP itself) [14, 15, 16]. And some are tailored to prominent mechanisms from the DP literature, such as differentially private stochastic gradient descent (DP-SGD), a cornerstone in privacy-preserving machine learning [17, 18, 19].

A common feature of all these prior works is that auditing DP for an algorithm is treated as a one-time task. In contrast, this work considers scenarios where a data processing mechanism is regularly monitored for potential privacy violations. The need for such ongoing auditing arises naturally in the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

context of privacy-preserving systems. First, the design of DP algorithms often involves extensive experimentation, making it valuable to track which modifications (inadvertently) result in privacy breaches. Second, during deployment, algorithms may undergo changes, typically aimed at improving efficiency or utility. And third, a trusted party responsible for validating the DP guarantees of these algorithms (such as a public agency or external auditor) might have an interest in continuously monitoring them over time.

In such situations, it is natural to model an algorithm as dynamically evolving over time. The task of a DP monitoring scheme would then be to continuously scan the algorithm for the emergence of potential privacy violations. In this context, an important insight is that monitoring cannot be feasibly done with traditional auditing schemes, say by applying them over and over again as a regular screening procedure. As we discuss in this work, such an approach puts at risk the reliability of individual screenings, and more importantly, is inefficient in its data use. We therefore present a new approach that conducts regular, data-efficient "small screenings" and aggregates the ensuing information over time. While any small screening may be neither particularly reliable nor sensitive, their combined (and carefully weighted) vote results in a powerful procedure that detects harmful changes shortly after they occur. The development of this procedure relies on state-of-the-art tools developed in probability theory to control the risk of false detections.

Main contributions In this work, we present, to the best of our knowledge, the first DP auditor that is specifically designed to monitor algorithms over extended periods of time. Our method comes with (1) significantly reduced sampling efforts, (2) tunable confidence parameters that contain the probability of false detection over the entire time span and (3) a mathematical theory that proves its soundness. In Section 2, we discuss basics of DP and sequential monitoring. We then present our monitoring method and accompanying statistical theory in Section 3. Subsequently, we evaluate our approach on common algorithms from the DP literature in Section 4 and conclude with a discussion of related works and limitations in Section 5.

2 Background

In this section, we briefly provide background on differential privacy and sequential monitoring. We point to Appendix B.2 for a brief introduction to stochastic processes.

Differential Privacy Roughly speaking, a (randomized) algorithm A is differentially private if the data of any one individual in the database x does not affect its output distribution by too much. More formally, DP implies that the output distribution of A on neighboring databases x and x' should be similar. Here, two datasets are neighbors if they differ in exactly one entry (i.e., one would obtain x' from x by altering the data in one row). We denote the neighborhood relation by $x \sim_N x'$.

Definition 1. *An algorithm A is ε-differentially private if for any two neighboring databases $x \sim_N x'$ and any measurable set $\mathcal{E}$*

$$\mathbb{P}\left(A(x) \in \mathcal{E}\right) \leq e^{\varepsilon}\, \mathbb{P}\left(A(x') \in \mathcal{E}\right). \tag{1}$$

If the privacy parameter $\varepsilon > 0$ is chosen small enough, an adversary observing A's output will not be able to confidently decide if database x or x' was used to produce said output. Therefore, assuming that each row of x corresponds to the data provided by one individual, an adversary cannot reliably infer membership in the dataset from the output.

To distinguish DP from its variants, Definition 1 is usually called *pure differential privacy*. The definition also serves as a useful basis for auditing the privacy of an algorithm under investigation. More concretely, prior works have relied on this definition to test the privacy claims of a given mechanism [8, 9, 20]. Vulnerable algorithms are found with the help of a suitable tuple $(x_0, x_0', \mathcal{E}_0)$ that provokes a privacy violation of the form

$$\mathbb{P}\left(A(x_0) \in \mathcal{E}_0\right) > e^{\varepsilon}\, \mathbb{P}\left(A(x_0') \in \mathcal{E}_0\right). \tag{2}$$

If, on the other hand, no violation as in (2) is observed for the tuple, then no violation of privacy is reported. A reliable auditing procedure will therefore also depend on a suitable choice of these tuples. While simple heuristics often suffice to find these [8], there also exist tools dedicated to their search [11]. We discuss suitable tuples for specific algorithms in our experiments in Section 4.

In our work, we also adopt an approach that leans on Definition 1 to check for privacy violations (see Section 3), with the difference that our auditor and privacy condition are formulated for sequential monitoring over time.

Sequential monitoring Sequential monitoring refers to the setting where an analyst observes data points $R_1, R_2, R_3, \ldots$ sequentially and looks for the emergence of a structural change over time. At any time τ, the analyst combines the available data $R_1, \ldots, R_\tau$ into a detector $\widehat{D}(\tau)$. When the next observation $R_{\tau+1}$ arrives, the detector is updated to $\widehat{D}(\tau+1)$, and so forth. A large detector value indicates that a change has taken place. More precisely, the analyst compares $\widehat{D}(\tau)$ to a threshold q and if $\widehat{D}(\tau) > q$, the analyst decides that a change has occurred and stops. Otherwise, if $\widehat{D}(\tau) \leq q$, no change is detected and monitoring continues until a terminal time T. A reliable monitoring method controls the false alarm rate (FAR), that is, the probability of incorrectly declaring a change when none has occurred. The FAR can be controlled by adjusting the threshold q, with larger values reducing the FAR but also making it harder to detect actual changes. It is common in the literature to set the FAR to an acceptable error level $\alpha \in (0, 1)$, typically 0.05. Since exactly obtaining FAR = α is difficult, most procedures only approximate this level for large T. For a detailed review of sequential monitoring, we refer the reader to [21]. The study of sequential monitoring (and more precisely the choice of an adequate threshold q) is closely connected to the theory of stochastic processes and to characteristics of the Brownian motion, discussed in Appendix B.2.

3 Privacy monitoring for an evolving algorithm

Overview In this section, we present a model for a randomized algorithm A that evolves over time. We treat time as a discrete variable, spanning $t = 1, 2, \ldots, T$, where each time point t may represent an hour, day, or week. Our focus is on real-time auditing of the differential privacy of A. Specifically, we consider the problem of continuously monitoring the outputs of A to detect privacy violations that may result from potential implementation bugs. Existing approaches to DP auditing are designed for fixed algorithms. A straightforward approach to monitoring A might involve repeatedly applying a standard DP-auditor at each time t across the entire time span. However, as we will demonstrate below, this naive approach is inefficient and leads to significant oversampling. To address this, we develop a new time-aggregator that carefully combines a sequence of naive auditors over time into a more powerful procedure. The methodology is detailed in Section 3.1. In Section 3.2, we present mathematical theory, providing performance guarantees for our approach. Our construction uses state-of-the-art tools from stochastic process theory that help to reduce sampling effort, while obtaining reliable decisions with regard to privacy.

3.1 Outline of the auditing methodology

Model for an evolving algorithm For our theoretical analysis, we consider an algorithm A that evolves over discrete time $t = 1, \ldots, T$. The output distributions of A may change with time, and we explicitly denote this dependence by writing A_t (the algorithm at time t). In our model, the algorithm may undergo modifications by the owner, and we denote the time points of these modifications by t_k, where $1 < t_1 < t_2 < \cdots < t_K < T$. Here, $K \in \mathbb{N}_0$ represents the total number of modifications, which may be zero. The relationship between these modifications can be expressed as follows:

$$A_1 = \cdots = A_{t_1-1} \neq A_{t_1} = \cdots = A_{t_2-1} \neq \cdots \neq A_{t_K} = \cdots = A_T. \tag{3}$$

In our theory, the analyst does not necessarily have to know the times or number of these modifications.

The NPVT condition Recall Definition 1 for the rigorous formulation of DP for a fixed algorithm. We now specify what it means for $(A_t)_{t=1,\cdots,T}$ to satisfy differential privacy (DP) over the entire time period. Consider a pair of neighboring databases $x \sim_N x'$ and a potentially vulnerable event $\mathcal{E}$, which represents a specific privacy risk. If A_t satisfies ε-DP at all times t, it must hold that

$$p_t := \mathbb{P}(A_t(x) \in \mathcal{E}) - e^{\varepsilon}\ \mathbb{P}(A_t(x') \in \mathcal{E}) \leq 0 \quad \forall t. \tag{4}$$

We can formulate this more compactly in the following "*no privacy violation over time*" (NPVT) condition

$$\text{NPVT:} \qquad \max_{1 \leq t \leq T} p_t \leq 0. \tag{5}$$

Scope and witness selection. Our procedure assumes access to repeated executions of each A_t on chosen neighboring databases x, x'. The tuple $(x, x', \mathcal{E})$ is selected before monitoring, using domain knowledge or preliminary simulations independent of the monitoring samples, and is then held fixed.

Finding informative tuples can be difficult in high dimensions and remains a separate task. Several preselected tuples can be monitored simultaneously using a Bonferroni correction; see Appendix A.4. Failure to detect a violation for the selected tuples does not certify DP over all neighboring databases and measurable events. Finally, NPVT concerns the privacy guarantee of each A_t separately; privacy of the joint sequence of released outputs requires a separate composition analysis.

The naive auditor We now address the task of monitoring A over time to detect implementation errors as they arise. Specifically, the analyst seeks to identify breaches of the NPVT condition in (5). At any time t, the analyst can execute the algorithm A_t with input databases x and x' to produce independent outputs. Denote these outputs as $X_{1,t}, \dots, X_{n,t} \sim A_t(x)$ and $Y_{1,t}, \dots, Y_{n,t} \sim A_t(x')$, where n is the number of runs. Using the datasets $\{X_{1,t}, \dots, X_{n,t}\}$ and $\{Y_{1,t}, \dots, Y_{n,t}\}$, the analyst can compute an empirical approximation of p_t as follows:

$$n_t^X := |\{1 \le i \le n : X_{i,t} \in \mathcal{E}\}|, \quad n_t^Y := |\{1 \le i \le n : Y_{i,t} \in \mathcal{E}\}| \quad \text{and} \quad \hat{p}_t := \frac{n_t^X - e^{\varepsilon} n_t^Y}{n}. \tag{6}$$

The counts n_t^X and n_t^Y have been used, for instance, in the seminal work of [8] to detect privacy violations. Moreover, $\hat{p}_t$ allows the construction of a confidence interval for p_t, forming the basis for an auditing procedure (all technical claims on the naive procedure are discussed in Appendix C.1). An accurate estimator of the variance of $\hat{p}_t$ is given by

$$\hat{\sigma}_t^2 := \frac{n_t^X}{n^2}\left(1 - \frac{n_t^X}{n}\right) + e^{2\varepsilon}\frac{n_t^Y}{n^2}\left(1 - \frac{n_t^Y}{n}\right). \tag{7}$$

Using the normal approximation of binomial random variables, we have

$$\frac{\hat{p}_t - p_t}{\hat{\sigma}_t} \approx \mathcal{N}(0,1), \tag{8}$$

where the right-hand side denotes a standard normal distribution. For a level $\alpha \in (0,1)$, a one-sided confidence interval for p_t is

$$\mathcal{C}_t(\alpha) := \big(\hat{p}_t + \Phi^{-1}(\alpha)\,\hat{\sigma}_t, \infty\big), \tag{9}$$

where Φ^{-1} is the quantile function of the standard normal. A privacy violation is detected if $0 \notin \mathcal{C}_t(\alpha)$. The logic of this decision is based on the fact that $p_t \in \mathcal{C}_t(\alpha)$ with high probability. If $0 \notin \mathcal{C}_t(\alpha)$, it must be because $0 < p_t$, indicating a breach of the DP condition (4). The auditor operates with an error level α, meaning it will only erroneously detect a violation with probability α if DP holds.
Having an auditor at a fixed time t, the question is how to scale it up to a full sequential auditor over all times $t = 1, \dots, T$. A first idea would be to apply the confidence intervals (9) at each time point and to declare a privacy violation as soon as $0 \notin \mathcal{C}_t(\alpha)$. This, however, does not control the probability of a false detection over the full monitoring period. Indeed, since fixed-time decisions are independent and each has false detection probability $\approx \alpha$, the probability of at least one false alarm over T audits is approximately $1 - (1-\alpha)^T$. For $\alpha = 5\%$ and $T = 50$, this is already larger than 90%. Thus, repeatedly applying a static auditor without adjustment is not a reliable monitoring strategy. This issue, often referred to as "alpha error accumulation", is shared by standard DP auditors when they are used sequentially; we illustrate this empirically in Appendix A.2 for the methods of [8] and [12].

A simple remedy is a Bonferroni correction: one applies the fixed-time auditor at level α/T at each time point, which ensures a global false detection probability of at most α. In our notation, this would mean declaring a DP violation at time t only if $0 \notin \mathcal{C}_t(\alpha/T)$. While valid, this approach is very conservative, especially for large T. More importantly, it still treats each time point separately and ignores the information contained in the previous samples $1, \dots, t-1$. Together, these issues contribute to low sensitivity against DP violations, and illustrations are given in our experiments. Our approach takes a different route. We construct a sequential statistic that aggregates evidence over time and is calibrated using stochastic-process limits. This yields global false alarm control and, most importantly, high sensitivity even against subtle privacy violations.

The weighted aggregator The naive auditor presented in the previous paragraph is based on the statistic $\hat{p}_t/\hat{\sigma}_t$. To see this, notice that no DP violation is detected if

$$0 \in \mathcal{C}_t(\alpha/T) \quad \Leftrightarrow \quad \frac{\hat{p}_t}{\hat{\sigma}_t} < -\Phi^{-1}(\alpha/T). \tag{10}$$

Conversely, a large enough value of $\hat{p}_t/\hat{\sigma}_t$ implies that $0 \notin \mathcal{C}_t(\alpha/T)$, indicating a DP violation. Our main idea is to combine these statistics over time to improve joint performance. The mathematical underpinnings of this analysis stem from the field of stochastic process theory, which lies at the heart of many modern statistical applications.

Recall that, at any time τ, the analyst has access to the historical sequence $\hat{p}_1/\hat{\sigma}_1, \hat{p}_2/\hat{\sigma}_2, \ldots, \hat{p}_\tau/\hat{\sigma}_\tau$. We propose the following aggregation scheme, parameterized by $\beta \in [0, 1/2)$ (for a recommended choice, see the end of this section):

$$\widehat{D}(\tau) := \max_{0 \le \ell < \tau} \left[\frac{1}{(\ell+1)^\beta \cdot T^{1/2-\beta}} \sum_{t=\tau-\ell}^{\tau} \frac{\hat{p}_t}{\hat{\sigma}_t} \right]. \tag{11}$$

A privacy violation is detected if $\widehat{D}(\tau)$ exceeds a critical threshold (details in the next paragraph). To understand $\widehat{D}(\tau)$, consider specific values of ℓ in (11). For $\ell = 0$, the aggregation reduces to

$$\frac{1}{(0+1)^\beta \cdot T^{1/2-\beta}} \sum_{t=\tau}^{\tau} \frac{\hat{p}_t}{\hat{\sigma}_t} = \frac{1}{T^{1/2-\beta}} \frac{\hat{p}_\tau}{\hat{\sigma}_\tau}. \tag{12}$$

Ignoring the scaling factor $T^{1/2-\beta}$, this is equivalent to the naive privacy estimator at time τ. The strength of this estimator is that it only relies on the most recent data. Its weakness is that it ignores potentially useful information from earlier times ($t < \tau$). If the right side of (12) is sufficiently large, a privacy violation is detected, and the procedure stops. Otherwise, the aggregator starts incorporating historical data by increasing ℓ. For $\ell = 1$, the aggregated statistic becomes

$$\frac{1}{(1+1)^\beta \cdot T^{1/2-\beta}} \sum_{t=\tau-1}^{\tau} \frac{\hat{p}_t}{\hat{\sigma}_t} = \frac{1}{2^\beta \cdot T^{1/2-\beta}} \left[\frac{\hat{p}_{\tau-1}}{\hat{\sigma}_{\tau-1}} + \frac{\hat{p}_\tau}{\hat{\sigma}_\tau} \right].$$

Thus, historical data from time $\tau - 1$ is added to see if this strengthens the evidence to conclude that a privacy violation has occurred. The data from time $\tau - 1$ potentially helps the auditor by reducing its variance. However, it is also not the latest data anymore - if a violation has only just occurred at time τ looking at the (private) history will not help at all. The ideal cut-off ℓ includes all recent helpful data while ignoring all prior, now obsolete, historical information. Since this "sweet spot" for ℓ is unknown, the procedure evaluates all possible values, combining them in $\widehat{D}(\tau)$. The final decision rule identifies $\widehat{D}(\tau)$ as unexpectedly large when it exceeds a predetermined threshold (indicating a privacy violation). The method to set this threshold is specified in the next paragraph.

Auditing via thresholding To summarize our auditing procedure briefly: At each time point τ, the user computes $\widehat{D}(\tau)$ and checks whether it exceeds a threshold q. If it does, a privacy violation is detected; otherwise, the auditing continues without detecting a violation. A formal description of this process is given in Algorithm 1. The choice of the threshold $q = q(\alpha)$ determines the FAR of the procedure (see Section 2), i.e., the risk α of falsely detecting a violation when none exists. If the naive procedure were used, the threshold would be $q(\alpha) = -\Phi^{-1}(\alpha/T)$ (see (10)). This is a quantile of the standard normal distribution. Similarly, for our aggregated statistic, $q(\alpha)$ is defined as the (upper) α-quantile of a more complex probability distribution involving the standard Brownian motion (sBm), a continuous-time stochastic process $B : [0, 1] \to \mathbb{R}$. An introduction to the sBm can be found in Section B.2. Using the sBm, we define the random variable

$$D := \sup_{0 \le v < u \le 1} d(u, v), \quad \text{where} \quad d(u, v) := \frac{B(u) - B(v)}{(u - v)^\beta}, \tag{13}$$

with parameter $\beta \in [0, 1/2)$ (the same parameter as in the definition of $\widehat{D}$ in (11)). Setting $q(\alpha)$ as the upper α-quantile of D provides an appropriate threshold to guarantee the confidence level α. In practice, one simply obtains $q(\alpha)$ from Monte Carlo simulations and details are given in Appendix B.4. The connection between $\widehat{D}$ and D is discussed in greater detail in the Appendix (see Lemma 1). Roughly speaking, $\widehat{D}$ can be expressed as maximizing the weighted increments of a stochastic process and it converges to the maximum of weighted increments of the sBm, i.e. to D. In practice, computing $q(\alpha)$ is feasible via Monte Carlo simulation. Importantly, this threshold can be determined prior to auditing since it does not depend on the specific output distributions of A_t. Our auditor in Algorithm 1 can be used as an extension of static tools, especially ones that also check for violations of DP with tuples $(x, x', \mathcal{E})$. And since our method, in principle, only requires access to the inputs

Algorithm 1 ReTiDP: Real-Time DP Auditing

Input: ε (targeted DP level), α (nominal level), β (weight parameter), n (sample size), T (time horizon), τ (current time), access to the inputs and outputs of $(A_t)_{1\leq t\leq \tau}$
Output: Auditing decision on DP violation
Compute $\hat{p}_t/\hat{\sigma}_t$ for all $1 \leq t \leq \tau$
Compute $\widehat{D}(\tau)$ using the formula defined in eq. (11)
Compute the upper α quantile $q(\alpha)$ of D
if $\widehat{D}(\tau) > q(\alpha)$ **then**
    Return "DP violation detected"
else
    Return "No DP violation detected"

and outputs of A, it is compatible with approaches that work in the white- [9], semi-black- [8] and black-box settings [11]. We conclude this section by giving additional details about our procedure.

Variance stabilization For certain (extreme) choices of $\mathcal{E}$ it may happen that the variance estimator $\hat{\sigma}_t^2$ in (7) is close to 0, which in turn may cause numerical instabilities. For this reason it can make sense to use the stabilized variance estimate

$$\tilde{\sigma}_t(c) := \max(\hat{\sigma}_t, c) \tag{14}$$

for some small, positive constant c. It turns out that our theoretical guarantees remain valid for suitable choices of c (see Corollary 1 below).

Weight parameter β Our procedure involves a user-determined weight parameter $\beta \in [0, 1/2)$ that occurs in the definition of $\widehat{D}$ and D. Roughly speaking, β implies a trade-off between finding larger privacy violations faster (β closer to $1/2$) and finding smaller violations more reliably (β closer to 0). A similar trade-off has been studied in statistical monitoring; see e.g. [22]. Our simulations suggest that choosing $\beta = 1/4$ right in the middle of the permissible range produces good overall results and we recommend this choice to users.

Extension to approximate differential privacy. The main focus of this work is methodological development for auditing pure DP. However, our methods also apply to approximate DP with minimal modifications. Recall that, for a fixed, known $\delta \in [0, 1)$, (ε, δ)-DP requires

$$\mathbb{P}(A_t(x) \in \mathcal{E}) \leq e^{\varepsilon}\mathbb{P}(A_t(x') \in \mathcal{E}) + \delta$$

for all neighboring databases x, x' and measurable events $\mathcal{E}$. For the chosen tuple $(x, x', \mathcal{E})$, the null condition therefore becomes $p_t \leq \delta$. Our monitoring procedure extends to this setting immediately by replacing $\hat{p}_t$ with $\hat{p}_t - \delta$ in (11). The standard-error estimator and the critical value $q(\alpha)$ remain unchanged. Indeed, subtracting δ leaves the variance and the centered estimation error unchanged:

$$(\hat{p}_t - \delta) - (p_t - \delta) = \hat{p}_t - p_t.$$

Under the same sampling and regularity assumptions, the false-alarm guarantee thus carries over to the null hypothesis $p_t \leq \delta$ for every $t = 1, \ldots, T$.

3.2 Theoretical analysis

Model specifications In this section we give mathematical performance guarantees for our monitoring scheme. For this purpose, recall the model (3) and the (to the analyst unknown) modification times $1 < t_1 < ... < t_K < T$. To formulate mathematical theory it is easiest to express the times of changes in terms of proportions of T and we hence set for some proportions $0 < \theta_1 < ... < \theta_K < 1$

$$t_i = \lfloor T\theta_i \rfloor, \quad i = 1, ..., K.$$

Here, the expression $\lfloor x \rfloor$ refers to the largest integer that is smaller than or equal to a number x (floor function). We now state the main theoretical assumption of our analysis.

Assumption 1. In our asymptotics, we let $n \to \infty$. The number of time points $T = T(n)$ may be increasing in n. Specifically, there exists a constant $\rho \in [0, 1)$, such that $T(n)/n^{\rho}$ converges to a positive real number as $n \to \infty$.

Assumption 1 is fairly mild. It essentially states that T needs to be asymptotically smaller order than n. This restriction is practically always fulfilled because DP auditing typically requires n to be much larger than T anyways (this will be true in all of our experiments). Also notice that the case where T is fixed and only $n \to \infty$ is covered by our theory.

Main results The following main mathematical result has two parts. First, it shows that if the NPVT condition holds (see eq. (5)), then the probability of false detection is bounded for large samples by α. The second part states that if NPVT is violated and there is a privacy violation, the probability of detecting this privacy violation goes to 1. In the results we let $n \to \infty$ but remember that $T = T(n)$ may be increasing as specified by Assumption 1.

Theorem 1. Suppose that Assumption 1 holds and denote by O the output of Algorithm 1 for some level $\alpha \in (0, 1)$.

i) If NPVT in (5) holds, it follows that $\limsup_{n\to\infty} \mathbb{P}\big(O = \text{"DP violation detected"}\big) \leq \alpha$.

ii) If NPVT in (5) does not hold, then $\lim_{n\to\infty} \mathbb{P}\big(O = \text{"DP violation detected"}\big) = 1$.

Part ii) of Theorem 1 states that privacy violations will be detected with asymptotic certainty (as n grows large). Next, we give a complementary result, where we show that the time between the first emergence of a persistent DP violation, called t_{k^*} and its detection, which we denote by τ^*, is very short.

Theorem 2. *Suppose that Assumption 1 holds and that the NPVT condition in (5) is violated. Then, it follows for the detection delay that* $\tau^* - t_{k^*} = \mathcal{O}_P\big(T^{(1/2-\beta)/(1-\beta)} \times n^{-1/(2(1-\beta))}\big)$.

We conclude this section with a small corollary on the variance stabilization.

Corollary 1. Suppose that Assumption 1 holds and let $(c_n)_n$ be a sequence of positive numbers such that $c_n \downarrow 0$. Then, the statements of Theorem 1 remain true, if one replaces $\hat{\sigma}_n$ in the definition of $\widehat{D}$ by $\tilde{\sigma}_n(c_n)$ (defined in eq. (14)).

4 Experiments

We evaluate our monitoring framework by applying it to algorithms from the DP literature that are known to be susceptible to privacy bugs. Our experiments show that our methods can quickly detect flawed variants and changes that result in a privacy breach.

4.1 Algorithms

Laplace Mechanism Additive noise algorithms like the Laplace mechanism can be used to privatize statistics and information extracted from a database. For instance, we can consider the sum statistic $S(x) = x_1 + x_2 + ... + x_m$ for a database $x = (x_1, ..., x_m)$ with $x_i \in [0, 1]$ for each $i \in \{1, ..., m\}$. The output $S(x)$ can then be privatized by adding Laplace noise calibrated to its sensitivity $\Delta_S := \max_{x \sim_N x'} |S(x) - S(x')|$. The sensitivity of S on the unit cube $[0, 1]^m$ is 1 and the Laplace mechanism, defined as

$$A(x) = S(x) + Z \tag{15}$$

for $Z \sim Lap(0, b)$, achieves a privacy level of $\varepsilon = \frac{1}{b}$. For our correct implementation of the Laplace mechanism, we choose $b = 1$, which yields $\varepsilon = 1$. We then consider two changes to A that adversely affect its privacy. The first is to choose a smaller noise scale $b = 0.95$, which only leads to a privacy level $\varepsilon' \approx 1.05$ instead of the targeted $\varepsilon = 1$. The second is to replace the Laplace noise in (15) entirely with noise from a Gaussian distribution that has the same mean and variance as the original perturbation term. In our setting, this means adding $Z \sim \mathcal{N}(0, 2b^2)$ to $S(x)$ with $b = 1$. Algorithm A in (15) would then not satisfy pure DP as in Definition 1, but merely approximate DP with parameters $\varepsilon = 1$ and $\delta \approx 0.0396$. To monitor the privacy of the Laplace mechanism, we choose neighboring databases $x = (0, 0, ..., 0)$ and $x' = (1, 0, ..., 0)$ with $m = 10$ and events of the form $\mathcal{E} = (-\infty, a]$ (we choose $a = 0$ for the change in noise scale and $a = -1$ for the change in noise type).

Report Noisy Max Many DP algorithms operate on queries Q_i evaluated on the database x, rather than the data itself. The Report Noisy Max algorithm (RNM), for instance, adds Laplace noise

$Z_i \sim Lap\big(0, \frac{2\Delta_{Q_i}}{\varepsilon}\big)$ to each query in $Q(x) = (Q_1(x), ..., Q_d(x))$ and returns the index of the largest noisy query answer, i.e.,

$$\arg\max_{i\in\{1,\cdots,d\}}(Q_1(x) + Z_1, ..., Q_d(x) + Z_d).$$

RNM achieves a privacy level of ε in this configuration. Note that there is an "information minimization" principle involved [23]. A single change in the database x could, in theory, affect each query Q_i and therefore inflate the associated privacy loss. However, since only the index of the largest noisy query is reported (as opposed to the entire noisy vector), the privacy of RNM remains stable. Accordingly, a flawed version of RNM studied in the literature [8] falsely reports the largest noisy value

$$\max_{i\in\{1,\cdots,d\}}(Q_1(x) + Z_1, ..., Q_d(x) + Z_d)$$

and thus breaks the promised privacy level ε. In our experiment, we study the change from the correct version of RNM (with parameter $\varepsilon = 1$) to this incorrect version. Similar to prior work, we focus on counting queries with sensitivity $\Delta_{Q_i} = 1$ and choose $Q(x) = (1, 1, 1, 1, 1)$ and $Q(x') = (2, 2, 2, 2, 2)$, as well as $\mathcal{E} = (-\infty, a]$ with $a = 2$ for our auditing procedure.

Sparse Vector Technique Another prominent algorithm that operates with noisy queries is the Sparse Vector Technique (SVT). Here, queries are again processed with Laplace noise and then checked against a noisy threshold. It is then reported if a noisy query lies above (output 1) or below (output 0) this threshold. The number of query answers above the threshold is limited by a bound $c \in \mathbb{N}$ and SVT stops processing queries once this limit is reached. In [6], several correct and erroneous versions of SVT are examined. We adopt the numbering from [6] while referring to these variants of SVT and choose SVT 2, a correct version that achieves its targeted privacy level ε, as a starting point. We assume that each query has the same sensitivity Δ and point to Algorithm 2 in Appendix A.1 for a more detailed outline of SVT 2. For our experiments, we undertake adjustments to SVT 2 that result in flawed versions of SVT, namely SVT 4, 5 and 6. We then test how fast our monitoring procedure detects these detrimental changes. More details on these changes can be found in Appendix A.1. We choose query patterns and events as in [8] for SVT 5 and SVT 6. A useful heuristic from [8] for selecting $\mathcal{E}$ is to choose the output we would obtain if we ran the algorithms without noise, which yields events of the form $\{Q(x)\}$. For SVT 4, we choose the same query pattern, but use the approach described in [12] to identify an output value that is associated with high privacy leakage. Said output will then be a likely candidate for detecting a privacy violation. By doing so, we select the event $\mathcal{E} = \{(0, 0, 0, 0, 0, 0, 1)\}$. An overview of the datasets and events used in our experiments can be found in Tables 1 and 2 in Appendix A.1.

In addition to these algorithms, we consider an example for DP-SGD and the extension of our technique to (ϵ, δ)-DP in Appendix A.1.

4.2 Setup and Plots

To illustrate our procedure, we consider various scenarios with harmful modifications. Experiments for the performance of our tool in the presence of harmless changes can be found in Appendix A.1. Throughout these experiments, we have fixed the time horizon at $T = 100$ and the sample size n at each time point at $n = 750$. Variations to the value of n are explored in Section A.3 of the appendix. In each scenario, a modification is introduced at time 50. We consider a total of 6 scenarios, where a list of the deployed algorithms before and after the change is given in Table 1 of Appendix A.1. Notice that the letters (a)-(f) in that table correspond to the labels in Figure 1. In each scenario, we consider 100 independent simulation runs to calculate $\{\widehat{D}(\tau) : 1 \leq \tau \leq T\}$ and compare them to the threshold $q(\alpha)$ for $\alpha = 0.05$ (details in Algorithm 1). If $\widehat{D}(\tau) > q(\alpha)$ we make the decision that "a privacy violation" has taken place and in Figure 1, we report at any time τ the proportion of runs where a privacy violation has already been detected. The vertical line at $\tau = 50$ in each plot marks the time at which the modification has actually taken place. The horizontal line at height 0.05 in each plot indicates the user-defined FAR α, which should not be exceeded in the absence of harmful changes. We compare the performance of our method (blue curve) against the StatDP hypothesis test from the seminal work in [8]. StatDP is described in more detail in Appendix A.2. It can be applied repeatedly, for each time point t, using the sample of n contemporaneous observations. As discussed after (9), the StatDP hypothesis test has to be applied with a Bonferroni correction, to avoid excessive false detections (see Appendix A.2 for an empirical documentation of this phenomenon). Moreover,

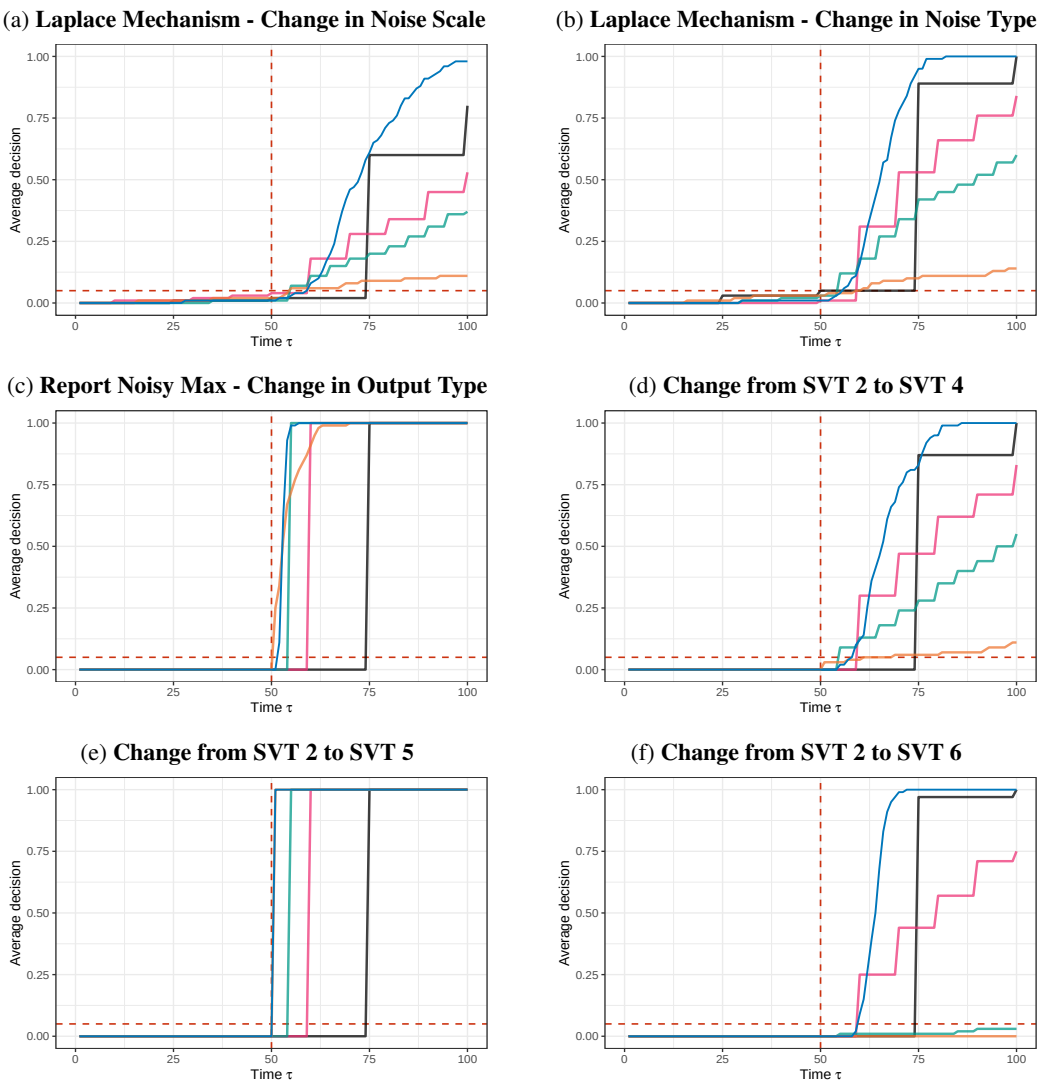


Figure 1: Average test decision of our detector ReTiDP (blue) and the batched version of StatDP [8] with step sizes $k = 1$ (orange), $k = 5$ (teal), $k = 10$ (magenta) and $k = 25$ (black). In panel (e), the curves for ReTiDP (blue) and StatDP with $k = 1$ coincide.

we add an equal-budget comparison in which the static auditor StatDP is applied only at every k-th time point, i.e., at times $k, 2k, \ldots, T$, using the $k \times n$ samples accumulated between each audit. This keeps the total sampling budget unchanged while trading greater power at each individual audit against lower temporal resolution. The significance level is adjusted via an appropriate Bonferroni correction in order to retain global FAR control across the resulting T/k audits. In our experiments, we choose $k = 1, 5, 10$ and 25.

Results Our experiments validate our procedure and illustrate key theoretical insights. First, for our new sequential monitoring scheme, we can see that in the absence of changes, the FAR is always upper bounded by α. In Figure 1 (a)-(f), the proportion of falsely detected violations for $\tau < 50$ is always lower than 0.05 (the horizontal line). This means that before any harmful modification occurs, the FAR is $\leq \alpha$. The FAR for the alternative methods is also well below α before a change occurs (due to the Bonferroni correction).

Next, we turn to times $\tau \geq 50$, when a privacy violation has occurred. Violations are detected by our new procedure with high certainty. Indeed, in all scenarios under consideration, the proportion of detected violations (blue curve) rises (close) to 100% in our experiments, showing that no violation

is entirely overlooked. The speed of detection, indicated by the gradient of the curves, varies, with stronger violations leading to faster detection.

In the immediate aftermath of the harmful change, the most favorable static choice of k can be (briefly) a bit more powerful than ReTiDP. However, ReTiDP quickly catches up and surpasses StatDP in nearly all scenarios. Moreover, we can observe that there is no uniformly best choice of k across times and violation types. Smaller values of k lead to more frequent testing. This tends to improve detection speed and is more suitable for detecting large DP violations quickly; however, it comes at the cost of lower overall detection power over the entire monitoring period. Larger k come with a longer detection delay but more statistical power overall, making them more suitable for detecting subtle DP violations. In practice, however, k must be chosen before knowing whether a violation will occur or not, and how large it will be. ReTiDP scans intervals adaptively: it reacts fast in both scenarios, while eventually attaining the highest power. No oracle knowledge about the size of the DP breach is needed.

5 Discussion

Related Work and Limitations Our work is conceptually close to previous approaches that audit algorithms with the help of tuples $(x, x', \mathcal{E})$ [8, 9, 11]. A comparison to these works also underlines the benefits of developing a method that is specifically designed for the monitoring task over extended periods of time. For instance, the auditors in [8] and [11] use up to $n = 5 \times 10^5$ and $n = 10.7 \times 10^6$ samples respectively to audit an algorithm once, and repeating this procedure over time would lead to high sampling costs. Our method works with $n = 750$ samples at each time t, a fraction of the samples used in these previous works.
While the use of tuples $(x, x', \mathcal{E})$ is a popular approach in auditing DP, it also comes with the necessity of choosing suitable databases and events. This can require specific knowledge of the DP mechanism under investigation or methods for identifying suitable events. To mitigate the dependence on a specific choice of tuples, one can apply the monitoring procedure simultaneously over different tuples (we provide an experiment in Appendix A.4). This approach alleviates but does not remove the problem entirely. Moreover, many related works rely on other tools than the tuples $(x, x', \mathcal{E})$ to audit DP. This is especially true for the various variants of DP, where privacy loss can oftentimes be expressed with the help of divergences. Accordingly, a number of approaches are built on these and other statistical distances rather than the tuples used in our approach [14, 16, 24, 13]. And many auditing approaches that are tailored to DP-SGD, an important DP mechanism for privacy-preserving machine learning, make use of f-differential privacy [18, 25, 26], another variant of DP that works without the aforementioned tuples. We believe that an interesting direction for future work would be to design a monitoring procedure compatible with these and other approaches.

Sequential testing versus monitoring. Sequential testing of a fixed mechanism and monitoring an evolving mechanism are distinct statistical problems. González Lara et al. [27] and Kutta et al. [28] develop sequential auditors for (ε, δ)-DP and f-DP, respectively. In their testing formulations, observations accumulate from a fixed mechanism, and the sample size adapts to the evidence against its privacy claim. Our monitoring problem allows the mechanism itself to change at unknown times. Its output distributions may change even under the null hypothesis, provided that privacy is preserved, and a violation may emerge, even for a mechanism that was initially safe. The objective is therefore to control false alarms throughout monitoring and to detect an emerging violation with a short delay measured from its unknown onset.

Conclusion We introduced a new problem: the continuous monitoring of a randomized algorithm for DP violations. This problem extends the scope of the DP auditing literature, which has traditionally treated auditing as a one-time task. For algorithms undergoing changes during development or deployment, consistent monitoring might be warranted to ensure sustained privacy. To address this, we proposed a novel method for aggregating a sequence of weak auditors into a reliable detector that provably controls false alarms. The aggregation mechanism employs a weighted majority vote over the weak auditors, assigning the greatest influence to the most recent observations. Our experiments demonstrate that this approach reliably and efficiently detects harmful changes across a variety of standard DP algorithms, while maintaining moderate sampling requirements for each screening.

Acknowledgments and Disclosure of Funding

Funding

This research received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through Germany's Excellence Strategy (EXC 2092 CASA, project number 390781972). Tim Kutta additionally acknowledges partial support from AUFF grants 47331 and 47222.

Competing interests

The authors declare no competing interests.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC'06*, 2006.

[2] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014.

[3] J. M. Abowd, "The U.S. census bureau adopts differential privacy," in *KDD'18*. ACM, 2018, p. 2867.

[4] N. Holohan, S. Braghin, P. M. Aonghusa, and K. Levacher, "Diffprivlib: The ibm differential privacy library," 2019.

[5] I. Mironov, "On significance of the least significant bits for differential privacy," in *the ACM Conference on Computer and Communications Security (CCS)*, 2012.

[6] M. Lyu, D. Su, and N. Li, "Understanding the sparse vector technique for differential privacy," *Proceedings of the VLDB Endowment*, vol. 10, no. 6, 2017.

[7] F. Tramèr, A. Terzis, T. Steinke, S. Song, M. Jagielski, and N. Carlini, "Debugging differential privacy: A case study for privacy auditing," 2022.

[8] Z. Ding, Y. Wang, G. Wang, D. Zhang, and D. Kifer, "Detecting violations of differential privacy," in *CCS'18*, 2018.

[9] B. Bichsel, T. Gehr, D. Drachsler-Cohen, P. Tsankov, and M. Vechev, "Dp-finder: Finding differential privacy violations by sampling and optimization," in *CCS'18*, 2018.

[10] Y. Wang, Z. Ding, G. Wang, D. Kifer, and D. Zhang, "Proving differential privacy with shadow execution," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, 2019.

[11] B. Bichsel, S. Steffen, I. Bogunovic, and M. T. Vechev, "Dp-sniper: Black-box discovery of differential privacy violations using classifiers," in *SP'21*, 2021.

[12] Ö. Askin, T. Kutta, and H. Dette, "Statistical quantification of differential privacy: A local approach," in *SP'22*, 2022.

[13] Y. Lu, M. Magdon-Ismail, Y. Wei, and V. Zikas, "Eureka: A general framework for black-box differential privacy estimators," in *SP'24*, 2024.

[14] X. Liu and S. Oh, "Minimax optimal estimation of approximate differential privacy on neighboring databases," in *NeurIPS'19*, 2019.

[15] J. Lokna, A. Paradis, D. I. Dimitrov, and M. T. Vechev, "Group and attack: Auditing differential privacy," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2023.

[16] T. Kutta, Ö. Askin, and M. Dunsche, "Lower bounds for rényi differential privacy in a black-box setting," in *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*.

[17] M. Jagielski, J. R. Ullman, and A. Oprea, "Auditing differentially private machine learning: How private is private sgd?" in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[18] M. Nasr, J. Hayes, T. Steinke, B. Balle, F. Tramèr, M. Jagielski, N. Carlini, and A. Terzis, "Tight auditing of differentially private machine learning," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023.

[19] M. Nasr, T. Steinke, B. Balle, C. A. Choquette-Choo, A. Ganesh, M. Jagielski, J. Hayes, A. G. Thakurta, A. D. Smith, and A. Terzis, "The last iterate advantage: Empirical auditing and principled heuristic analysis of differentially private SGD," in *ICLR 2025*.

[20] Y. Wang, Z. Ding, D. Kifer, and D. Zhang, "Checkdp: An automated and integrated approach for proving differential privacy or finding precise counterexamples," in *CCS'20*, 2020.

[21] A. Aue and C. Kirch, "The state of cumulative sum sequential change point testing seventy years after Page," *Biometrika*, 2024.

[22] T. Kutta and N. Dörnemann, "Monitoring time series with short detection delay," *Electron. J. Statist.*, vol. 19, no. 1, pp. 2239–2275, 2025.

[23] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[24] W. Kong, A. M. Medina, M. Ribero, and U. Syed, "Dp-auditorium: a large scale library for auditing differential privacy," in *45th IEEE Symposium on Security and Privacy (SP)*, 2024.

[25] M. S. M. S. Annamalai, B. Balle, E. D. Cristofaro, and J. Hayes, "To shuffle or not to shuffle: Auditing DP-SGD with shuffling," *arXiv:2411.10614*, 2024.

[26] M. S. M. S. Annamalai and E. D. Cristofaro, "Nearly tight black-box auditing of differentially private machine learning," *arXiv:2405.14106*, 2024.

[27] T. González Lara, M. Dulce Rubio, A. Ramdas, and M. Ribero, "Sequentially auditing differential privacy," in *Advances in Neural Information Processing Systems*, vol. 38, 2025.

[28] T. Kutta, M. Dunsche, Y. Wei, and V. Zikas, "Sequential auditing for f-differential privacy," in *35th USENIX Security Symposium (USENIX Security 26)*, 2026.

[29] Ö. Askin, H. Dette, M. Dunsche, T. Kutta, Y. Lu, Y. Wei, and V. Zikas, "General-purpose f-dp estimation and auditing in a black-box setting," in *34th USENIX Security Symposium, USENIX Security*, 2025.

[30] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 84, no. 1, pp. 3–37, 2022.

[31] C.-S. J. Chu, M. Stinchcombe, and H. White, "Monitoring structural change," *Econometrica*, vol. 64, pp. 1045–1065, 1996.

[32] P. Qiu, *Introduction to Statistical Process Control*. Boca Raton, FL: Chapman and Hall/CRC, 2014.

[33] L. Horváth, M. Hušková, P. Kokoszka, and J. Steinebach, "Monitoring changes in linear models," *Journal of Statistical Planning and Inference*, vol. 126, pp. 225–251, 2003.

[34] L. Horváth, P. Kokoszka, and S. Wang, "Monitoring for a change point in a sequence of distributions," *The Annals of Statistics*, vol. 49, pp. 2271–2291, 2021.

[35] T. Kutta, A. Jach, and P. Kokoszka, "Monitoring panels of sparse functional data," *J. Time Ser. Anal.*, 2025.

[36] A. Aue and L. Horváth, "Delay time in sequential detection of change," *Statistics & Probability Letters*, vol. 67, no. 3, pp. 221–231, 2004.

[37] Y. Yu, O. H. M. Padilla, D. Wang, and A. Rinaldo, "A note on online change point detection," *Sequential Analysis*, vol. 42, no. 4, pp. 438–471, 2023.

[38] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*, 2nd ed., ser. Graduate Texts in Mathematics. New York, NY: Springer, 1991, vol. 113.

[39] A. W. van der Vaart and J. A. Wellner, *Weak Convergence and Empirical Processes. With Applications to Statistics*. New York: Springer Series in Statistics., 1996.

[40] A. Račkauskas and C. Suquet, "Hölderian invariance principle for hilbertian linear processes," *ESAIM: Probability and Statistics*, vol. 13, pp. 261–275, 2009.

[41] P. Billingsley, *Convergence of Probability Measures*, 2nd ed., ser. Probability and Statistics. John Wiley & Sons, 1999.

[42] T. Kutta and P. Kokoszka, "Prokhorov metric convergence of the partial sum process for reconstructed functional data," *Electronic Journal of Statistics*, vol. 20, no. 2, pp. 3908–3958, 2026.

[43] H. Berbee, "Convergence rates in the strong law for bounded mixing sequences," *Probability Theory and Related Fields*, vol. 74, no. 2, pp. 255–270, 1987.

A Additional simulations and further details

In this section, we give additional details on the experimental setup and report further simulations that illuminate different aspects of our monitoring procedure. All experiments are implemented in R and run on a local device with an Apple M3 Pro chip and 18 GB of memory.

A.1 Additional mechanisms and details

Below, in Algorithm 2, we give the pseudo code for SVT 2, as discussed in [6] and used in our experiments. SVT 2 is a correct version of the sparse vector technique that achieves its targeted privacy level ε. We use it as a starting point in our experiments and simulate privacy violations by making changes to SVT 2 that result in faulty versions of the sparse vector technique.
One flawed modification of SVT 2 encountered in the literature is SVT 4. In this variant, the calibration of the noise in steps 2 and 4 of Algorithm 2 is done without taking the bound c into account, and the resampling of the threshold noise in step 7 is omitted. Other incorrect variants neglect to add noise to the queries $Q_i(x)$ (SVT 5) or fail to bound the number of noisy queries that lie above the noisy threshold (SVT 6). We point to [6] for more details on these flawed variants of SVT. Throughout our experiments, we set $\varepsilon = 1$, $\Gamma = 1$ and $c = 1$.

Algorithm 2 SVT 2 (from [6])

Input: x (database), $Q_1, ..., Q_d$ (queries), Γ (threshold), ε (target privacy level), $c \in \mathbb{N}$ (bound)
Output: $y_1, y_2, \cdots$ (binary sequence)

1: Set count $\leftarrow 0$
2: Sample $Z \sim Lap(0, 2c\,\Delta/\varepsilon)$
3: **for** $i = 1, ..., d$ **do**
4: Sample Z_i with $Z_i \sim Lap(0, 4c\,\Delta/\varepsilon)$
5: **if** $Q_i(x) + Z_i \geq \Gamma + Z$ **then**
6: **Output** $y_i = 1$
7: Sample $Z \sim Lap(0, 2c\,\Delta/\varepsilon)$
8: Set count $\leftarrow$ count + 1
9: **if** count $\geq c$ **then**
10: **Abort**
11: **else**
12: **Output** $y_i = 0$

Besides SVT, we consider the DP-SGD instance of [29] (Section 6.1 and Appendix B.1): noisy gradient descent for mean estimation with loss $\ell(\theta, x_i) = \frac{1}{2}(\theta - x_i)^2$ on $\Theta = \mathbb{R}$ (so no projection is needed), starting from $\theta_0 = 0$. In each of $K = 10$ iterations, a batch $I_k \subseteq \{1, \dots, r\}$ of $m = 5$ of the $r = 10$ records is drawn uniformly at random, and

$$\theta_k = \theta_{k-1} - \rho\Big(\frac{1}{m}\sum_{i \in I_k}(\theta_{k-1} - x_i) + Z_k\Big), \qquad Z_k \sim \mathcal{N}(0, \sigma^2),$$

is computed with learning rate $\rho = 0.2$ and noise level $\sigma = 0.2$. The output is θ_K. For the neighboring databases $x = (0, \dots, 0)$ and $x' = (1, 0, \dots, 0)$, the exact trade-off curve within the f-DP framework

$$T(A(x), A(x'))(\alpha) = \sum_{I \subseteq \{1,\dots,K\}} 2^{-K}\, \Phi\big(\Phi^{-1}(1-\alpha) - \mu_I/\bar{\sigma}\big)$$

can be calculated with $\bar{\sigma}^2 = \rho^2\sigma^2\big(1-(1-\rho)^{2K}\big)/\big(1-(1-\rho)^2\big)$ and $\mu_I = \sum_{k \in I}\rho(1-\rho)^{K-k}/m$.

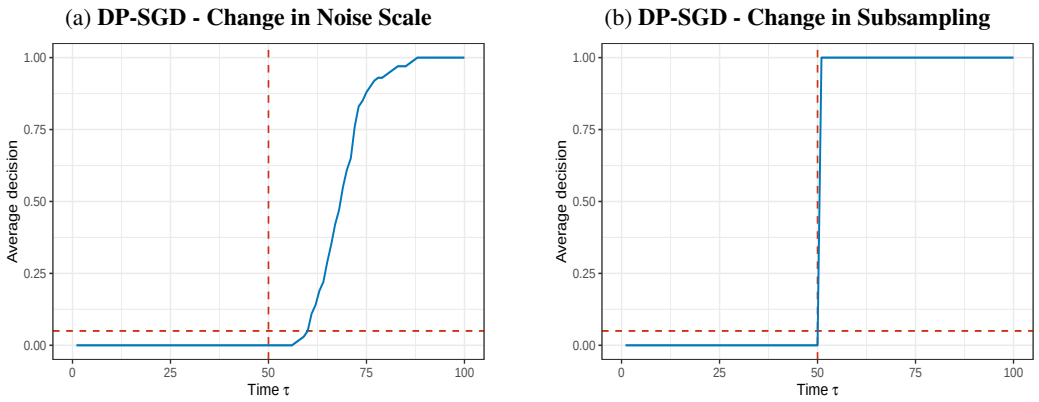


Figure 2: Average test decision of our detector ReTiDP (blue) applied to the flawed examples of DP-SGD. StatDP is designed for examining pure ε-DP and is therefore not applied here.

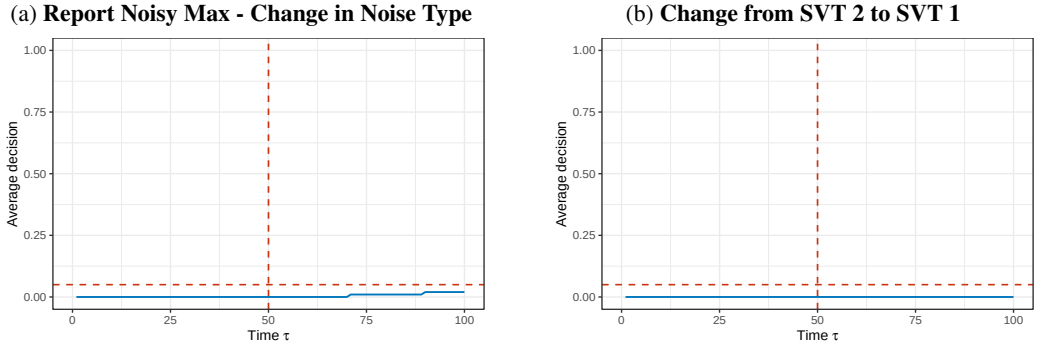


Figure 3: Average test decision of ReTiDP (blue). A benign modification happens at time 50 and does not introduce a privacy violation.

We can now use this exact trade-off from [29] and the primal–dual relation between f-DP and (ε, δ)-DP (Proposition 2.12 in [30]) to obtain the optimal (ε, δ)-guarantees for this instance of DP-SGD. With $A(x) \sim P$ and $A(x') \sim Q$, the smallest δ for a given ε is obtained from the convex conjugate of the trade-off function, $\delta(\varepsilon) = \max\{1 + T(P,Q)^*(-e^\varepsilon),\ 1 + T(Q,P)^*(-e^\varepsilon)\}$, taking both orders of P and Q. For $\varepsilon = 1$ this gives $\delta = 0.264$, i.e. the mechanism satisfies $(1, 0.264)$-DP for this pair of databases.

We monitor the claimed $(1, 0.264)$-DP guarantee with ReTiDP, replacing $\hat{p}_t$ by $\hat{p}_t - \delta$, on the databases x, x' and the event $\mathcal{E} = (-\infty, 0]$, with $n = 750$ samples per database and time point, $T = 100$ and level $\alpha = 0.05$. The mechanism changes after $t = 50$ are (a) a reduction of the noise level by 20 % from $\sigma = 0.2$ to $\sigma = 0.16$ and (b) the introduction of a bug into the subsampling subroutine, with the batch always consisting of the first $m = 5$ records instead of being chosen randomly. Both changes violate the claimed privacy guarantee and ReTiDP detects both breaches reliably (see Figure 2).

Benign changes

In the main experiment section, we focus on harmful modifications and study whether DP violations are detected quickly and reliably. However, not every modification of an algorithm is harmful. In particular, a modification from one ε-DP algorithm to another ε-DP algorithm should not be flagged as a privacy violation. We therefore also investigate benign changes and verify that our monitoring procedure remains stable in these cases.

We consider two examples already connected to the experimental setup in Section 4.2. First, applying exponential noise $Exp(2/\varepsilon)$ to the counting queries in Report Noisy Max should not affect its privacy guarantee [8]. Second, modifying SVT 2 in a way that yields SVT 1 from [6] should also preserve the privacy parameter ε. We apply our monitoring procedure to these benign changes using the

same parameters and query patterns as before (with the exception of $c = 2$ for the SVT algorithms). The corresponding choices of databases and events are listed in Tables 1 and 2. Detection rates are reported in Figure 3. In both scenarios, the detection rate remains low before and after the modification. This is the intended behavior: the procedure is sensitive to harmful changes, but does not react systematically to modifications that preserve the targeted DP guarantee.

Finally, we provide the full list of algorithms, databases and events used in our auditing experiments in Tables 1 and 2. These also include those used in our examples for DP-SGD and the modifications that do not cause privacy violations and are therefore benign.

Table 1: Summary of the algorithms and scenarios investigated in our simulations.

Scenario	**Algorithm**	**Modification**
Figure 1 (a)	Laplace mechanism	Decrease Noise Scale
Figure 1 (b)	Laplace mechanism	Change Noise Type to Gaussian
Figure 1 (c)	Report Noisy Max	Report maximum noisy query
Figure 1 (d)	SVT 2	Change to SVT 4
Figure 1 (e)	SVT 2	Change to SVT 5
Figure 1 (f)	SVT 2	Change to SVT 6
Figure 2 (a)	DP-SGD	Decrease Noise Scale
Figure 2 (b)	DP-SGD	Flawed Subsampling
Figure 3 (a)	Report Noisy Max	Change Noise Type to Exponential
Figure 3 (b)	SVT 2	Change to SVT 1

Table 2: Summary of the databases and events used in our different scenarios.

	Database x or query $Q(x)$	**Database x' or query $Q(x')$**	**Event $\mathcal{E}$**
Fig 1 (a)	$(0,0,0,0,0,0,0,0,0,0)$	$(1,0,0,0,0,0,0,0,0,0)$	$(-\infty, 0]$
Fig 1 (b)	$(0,0,0,0,0,0,0,0,0,0)$	$(1,0,0,0,0,0,0,0,0,0)$	$(-\infty, -1]$
Fig 1 (c)	$(1,1,1,1,1)$	$(2,2,2,2,2)$	$(-\infty, 2]$
Fig 1 (d)	$(0,0,0,0,0,1,1,1,1,1)$	$(1,1,1,1,1,0,0,0,0,0)$	{(0,0,0,0,0,0,1)}
Fig 1 (e)	$(0,0,0,0,0,1,1,1,1,1)$	$(1,1,1,1,1,0,0,0,0,0)$	{Q(x)}
Fig 1 (f)	$(1,1,1,1,1,0,0,0,0,0)$	$(0,0,0,0,0,1,1,1,1,1)$	{Q(x)}
Fig 2 (a)	$(0,0,0,0,0,0,0,0,0,0)$	$(1,0,0,0,0,0,0,0,0,0)$	$(-\infty, 0]$
Fig 2 (b)	$(0,0,0,0,0,0,0,0,0,0)$	$(1,0,0,0,0,0,0,0,0,0)$	$(-\infty, 0]$
Fig 3 (a)	$(1,1,1,1,1)$	$(2,2,2,2,0)$	$\{5\}$
Fig 3 (b)	$(0,0,0,0,0,1,1,1,1,1)$	$(1,1,1,1,1,0,0,0,0,0)$	{(0,0,0,0,0,1,1)}

A.2 The StatDP and DPL Algorithms

Our monitoring procedure is conceptually closest to the StatDP tool from [8]. Here, the privacy condition in (4) is treated as a testing problem with

$$H_0 : \mathbb{P}(A_t(x) \in \mathcal{E}) \leq e^{\varepsilon}\ \mathbb{P}(A_t(x') \in \mathcal{E}) \quad \text{vs.} \quad H_1 : \mathbb{P}(A_t(x) \in \mathcal{E}) > e^{\varepsilon}\ \mathbb{P}(A_t(x') \in \mathcal{E})$$

at each time t. A hypothesis test for the above problem can be constructed with the help of the empirical probabilities n_t^X and n_t^Y in equation (6). More concretely, [8] make use of Fisher's exact test to derive a p-value for the above testing problem, namely

$$\rho = 1 - F_{Hyper}(\tilde{n}_t^X - 1 | 2n, n, \tilde{n}_t^X + n_t^Y),$$

where F_{Hyper} is the cumulative distribution function of the Hypergeometric distribution and $\tilde{n}_t^X \sim Bin(n_t^X, 1/e^{\varepsilon})$. A privacy violation is detected at time t if $\rho \leq \alpha$, i.e., the hypothesis H_0 is rejected

(a) **Error Accumulation for StatDP** (b) **Error Accumulation for DPL**

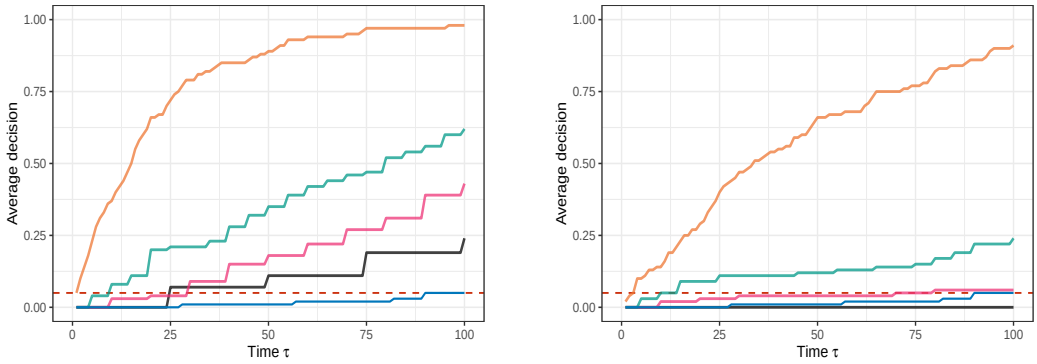


Figure 4: Error accumulation for ReTiDP (blue) and the batched versions of StatDP and DPL without the Bonferroni correction and with $k = 1$ (orange), $k = 5$ (teal), $k = 10$ (magenta) and $k = 25$ (black). The data are generated by the standard Laplace mechanism, with no modification over time and no violation of the targeted DP level.

in favor of the alternative H_1. To keep an overall level α, StatDP is applied with a Bonferroni correction in our simulations. Moreover, we run StatDP with the same events and databases we use for our own monitoring procedure.

In addition to StatDP, we can compare our monitoring tool against a pointwise adaptation of the local privacy loss bound of [12], applied to a preselected tuple. The DPL method from [12] is designed to audit the privacy claims of algorithms with the help of the empirical privacy loss

$$\hat{\ell}_{x,x'}(y^*) = |\ln(\tilde{p}_x(y^*)) - \ln(\tilde{q}_{x'}(y^*))|,$$

where $\tilde{p}_x$ and $\tilde{q}_{x'}$ are floored density estimators for the densities of $A(x)$ and $A(x')$ respectively, while y^* is the point where $\hat{\ell}$ achieves its maximum over a compact subset C. The privacy loss can be used to derive a statistical lower bound

$$LB := \hat{\ell}_{x,x'}(y^*) + \frac{\Phi^{-1}(\alpha)\,\hat{s}}{c_n}$$

for the targeted privacy parameter ε with $\mathbb{P}(LB \leq \varepsilon) \gtrsim 1 - \alpha$. Here, Φ^{-1} is the quantile function of the standard normal, c_n a normalizing constant and $\hat{s}^2$ a variance estimator. More details on the lower bound LB and these terms can be found in Section 4 of [12]. We can use the DPL method to monitor our algorithms for privacy breaches by computing the lower bound LB_t anew at each time step t. A privacy violation at time t is then detected if we observe $LB_t > \varepsilon$. In our simulations, we use the discrete and kernel density estimators from [12] with the same specifications and a floor equal to 10^{-3} for the density estimators. We then compute the lower bound LB_t as in [12] and compare it to the targeted privacy level ε at each time t to decide if a privacy breach has occurred. Moreover, we can run a batched version of the DPL method similar to StatDP (with step sizes $k = 1$, $k = 5$, $k = 10$ and $k = 25$).

Alpha error accumulation

In Section 3.1, after (9), we argue that one-off DP auditors cannot be turned into monitoring procedures by simply repeating them over time. The reason is that their false detection rate is calibrated for a single decision, not for a sequence of decisions. When such an auditor is run repeatedly, the probability of observing at least one false alarm over the monitoring period can become much larger than the target level α.

We illustrate this effect empirically for the auditors of [8] and [12]. The experiment is deliberately simple. We use the standard Laplace mechanism, calibrated to satisfy the targeted ε-DP guarantee, and monitor it for up to $T = 100$ time points. The mechanism is not modified during the experiment. Hence, the NPVT condition holds by construction and every reported privacy violation is necessarily a false detection. At each time point, we apply the corresponding one-off auditor at the conventional level $\alpha = 5\%$, without Bonferroni correction or any other adjustment for repeated testing. In

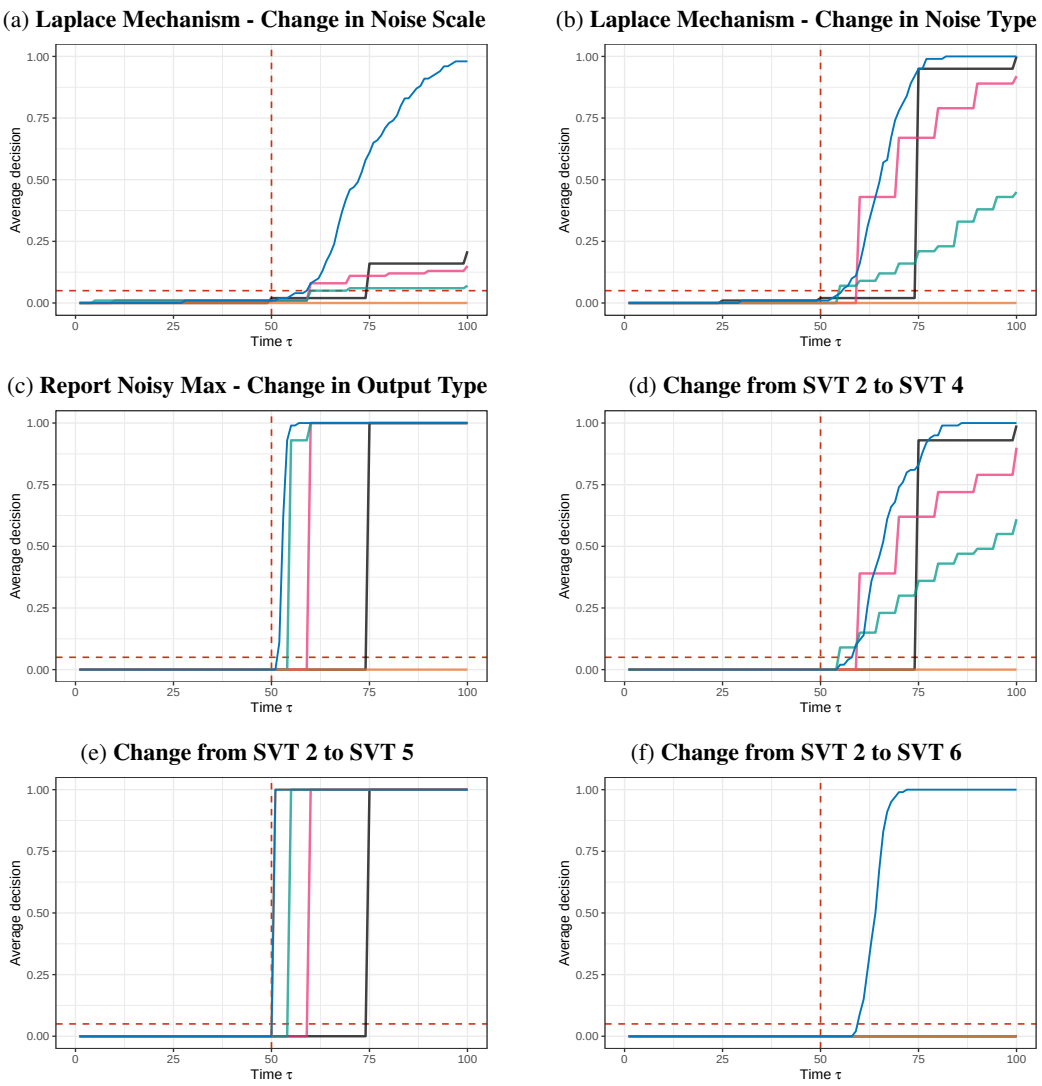


Figure 5: Average test decision of our detector ReTiDP (blue) and the batched version of DPL [12] with step sizes $k = 1$ (orange), $k = 5$ (teal), $k = 10$ (magenta) and $k = 25$ (black). In panel (e), the curves for ReTiDP (blue) and DPL with $k = 1$ coincide. In panel (f), the curves for DPL with $k = 1$ (orange) and the other step sizes $(k = 5, 10, 25)$ coincide.

addition, we run the batched versions of StatDP and DPL with step size $k = 5, 10$ and 25 without the appropriate Bonferroni corrections. We also add our ReTiDP method and compare its performance against both baselines.

Figure 4 reports the empirical probability that at least one privacy violation has been declared by time τ. The results show the accumulation effect clearly: although the mechanism remains private throughout, the false detection probability for both StatDP and DPL increases visibly with the monitoring horizon and becomes close to one by $T = 100$ for $k = 1$. Thus, an uncorrected sequence of static audits can produce apparent privacy violations even for the same unchanged private algorithm. Our monitoring procedure, on the other hand, keeps the FAR below or close to α. For step sizes $k = 10$ and $k = 25$, the DPL method also manages a FAR below or around α. This is however achieved at the cost of detection speed. In fact, our experiments below indicate that the lower bound for the DPL method in our setting is somewhat conservative.

In summary, this is precisely the failure mode that motivates monitoring procedures with global false alarm control.

(a) **Laplace Mechanism - Change in Noise Type** (b) **Change from SVT 2 to SVT 4**

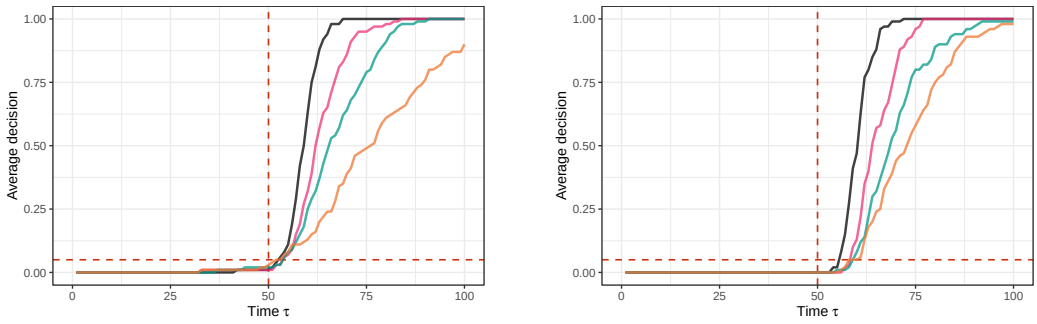


Figure 6: Scenarios (b) and (d) from Figure 1 for $n = 200$ (orange), 500 (teal), 1000 (magenta), and 2000 (black).

Flawed Algorithms and DPL

To compare our adaptation of the DPL method against ReTiDP, we deploy it on the same databases as our monitoring procedure and use the same events for auditing with the exception of the events of the form $\mathcal{E} = (-\infty, a]$. The reason for this is that the loss function in DPL is computed on single output values rather than on intervals. Thus, for these events (see $\mathcal{E}$ in Fig 1 (a)-(c) from Table 2) we use $y^* = -0.75$, $y^* = -2.5$ and $y^* = 1.25$ respectively to compute the empirical loss and lower bound.

Our experiments in Figure 5 show that the detection speed and performance of DPL rarely match that of ReTiDP. In most cases, ReTiDP performs significantly better than the DPL method and the plots confirm that DPL is a rather conservative detector that sacrifices some power for a higher degree of confidence.

A.3 The impact of n

We study the effect of the sample size n available at each time point. For this purpose, we revisit scenarios (b) and (d) from Section 4.2. All experimental specifications are kept fixed, except that we repeat the experiments with $n = 200, 500, 1000, 2000$. The results are displayed in Figure 6. Before the modification at time $\tau = 50$, the FAR remains below 0.05 for all choices of n and for both algorithms, in line with Theorem 1. After the modification, larger values of n lead to faster detection. This is consistent with Theorem 2, which predicts that the detection delay decreases as the available sample size increases. In both scenarios, the curves rise more quickly for larger n, confirming the expected gain in sensitivity from additional samples.

A.4 Monitoring with multiple witnesses

In practice, it can be advisable to monitor an algorithm for several different configurations at the same time, since the choice of algorithm parameters, databases and events can influence how well privacy violations are detected. In that case, we need to aggregate the information from all monitoring runs into one decision for each time step τ. We can do so by deciding that a privacy violation has occurred as soon as at least one of our monitoring runs detects a privacy violation. In order to not inflate the FAR, we apply a Bonferroni correction across all configurations.

Remark (Details on multiple monitoring). ReTiDP can monitor a finite (fixed) collection of witnesses $\mathcal{W} = \{(x_w, x'_w, E_w) : w = 1, \dots, m\}$, where $x_w \sim_N x'_w$. Suppose that the assumptions of Theorems 1 and 2 hold for each witness. Run each monitor at level α/m, using threshold $q(\alpha/m)$, and declare a privacy violation if any monitor rejects. Under NPVT for all witnesses, Theorem 1 and the union bound give

$$\limsup_{n\to\infty} \mathbb{P}(\text{at least one false detection by time } T) \leq \sum_{w=1}^{m} \frac{\alpha}{m} = \alpha.$$

This controls false alarms jointly over time and witnesses.

Table 3: Witnesses for multiple monitoring.

Mechanism	J	Δ_s	Δ_a	$a_1, \ldots, a_J$
Laplace, $b = 1 \to 0.85$	5	0.25	0.75	$-1.5, \ldots, 1.5$
	10	0.1	0.3	$-1.35, \ldots, 1.35$
	20	0.05	0.15	$-1.425, \ldots, 1.425$
DP-SGD, $\sigma = 0.2 \to 0.15$	5	0.25	0.025	$0, \ldots, 0.1$
	10	0.1	0.01	$0, \ldots, 0.09$
	20	0.05	0.005	$0, \ldots, 0.095$

The level correction increases the critical value, but only slowly. The central reason is that suprema of Gaussian processes strongly concentrate, so their survival function has a very light tail. Let us make this more precise. For $0 \le \beta < 1/2$, it holds that

$$\mathrm{Var}\left(\frac{B(u) - B(v)}{(u-v)^\beta}\right) = (u-v)^{1-2\beta} \le 1, \qquad 0 \le v < u \le 1.$$

Then, the famous Borell–TIS inequality implies square-exponential tails, i.e. for some $c_1, c_2 > 0$ (depending only on β)

$$\mathbb{P}(D > x) \le c_1 e^{-c_2 x^2}, \qquad x \ge 0.$$

Thus, $q(\eta) = O(\sqrt{\log(1/\eta)})$ as $\eta \downarrow 0$.

Let us look at the effect for change point testing. Following the proof of Theorem 2, we see that delays are of scale

$$\left\{\frac{q(\alpha/m)\, T^{1/2-\beta}}{\sqrt{n}}\right\}^{1/(1-\beta)}.$$

But then for large m (many events)

$$q(\alpha/m)^{1/(1-\beta)} = O\Big(\{\log(m/\alpha)\}^{1/[2(1-\beta)]}\Big),$$

which has exponent $2/3$ for $\beta = 1/4$. Relative to monitoring this witness alone, the corresponding delay scales differ by the factor $\{q(\alpha/m)/q(\alpha)\}^{1/(1-\beta)}$, which is quite modest; so the penalty for multiple monitoring is relatively mild (this we also see in our simulations). Moreover, and on the theoretical side: for any fixed finite panel, the consistency conclusion of Theorem 1 remains valid whenever at least one witness has a privacy violation.

Experiments

For our experiments, we monitor two mechanisms on $J \in \{5, 10, 20\}$ witnesses simultaneously.

We deploy the Laplace mechanism from before on databases $x \in [0,1]^{10}$. After $t = 50$ the noise scale drops by 15% from $b = 1$ to $b = 0.85$, so the mechanism is only $1/0.85 \approx 1.18$-DP and violates the claimed 1-DP guarantee. Additionally, we deploy DP-SGD as in Appendix A.1, with the claimed $(1, 0.264)$-DP guarantee for $\sigma = 0.2$. After $t = 50$ the noise level drops by 25% to $\sigma = 0.15$.

Witness $j = 1, \ldots, J$ compares the databases

$$x_j = (0, s_j, 0, \ldots, 0), \qquad x_j' = (1, s_j, 0, \ldots, 0) \tag{16}$$

on the event $E_j = (-\infty, a_j]$, where

$$s_j = (j-1)\,\Delta_s, \qquad a_j = a_1 + (j-1)\,\Delta_a. \tag{17}$$

Every pair differs by 1 in the first record. The first witness consists of the original databases $(0, \ldots, 0)$ and $(1, 0, \ldots, 0)$. The steps Δ_s and Δ_a are given in Table 3. Each witness is monitored with ReTiDP on its own samples, $n = 750$ per database and time point, with $T = 100$. To control the overall level, each witness is tested at level α/J (Bonferroni). The aggregate decision raises an alarm as soon as

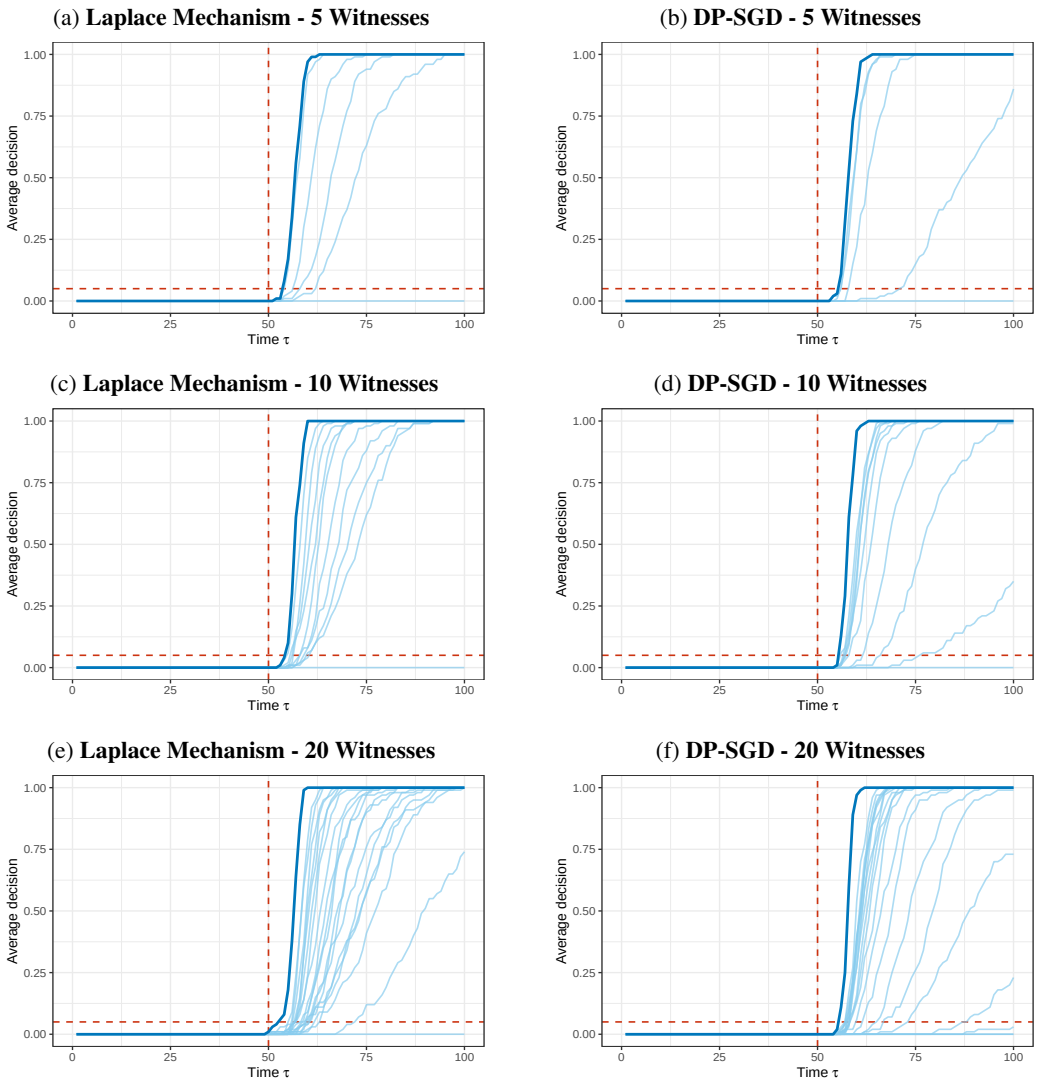


Figure 7: Light blue: average decision of ReTiDP on each witness, at level α/J; overlapping curves appear darker. Blue: aggregate decision, which raises an alarm as soon as any witness does.

any witness does, so its false-alarm rate is at most $\alpha = 0.05$. The threshold $q(\alpha/J)$ lies further in the tail than $q(\alpha)$ and is estimated from 20,000 instead of 5,000 simulated draws. The results of our simulation can be observed in Figure 7.

Our empirical results, reported in Figure 7, are compatible with our theory. We observe that, due to the error corrections, pre-change rejection probabilities always stay well below the nominal level of 5%. After the change, we observe a rapid increase in rejections, with the aggregate decision (bold line) mechanically more powerful than each of the individual decisions (light lines). The most important observation is that some events with low vulnerability do not prevent detection of privacy violations through the resulting error corrections.

A.5 Finite-sample calibration.

Our theoretical false-alarm guarantees are asymptotic. We examine their finite-sample accuracy using the corresponding asymptotic critical values. For $\varepsilon = 1$, we generate mutually independent $B^X_{i,t} \sim \text{Bernoulli}(a)$ and $B^Y_{i,t} \sim \text{Bernoulli}(e^{-1}a)$, $a \in \{0.30, 0.50\}$, so that the privacy contrast equals zero at every time. We use $n \in \{200, 600, 1000, 1400\}$, $T \in \{50, 100, 150, 200\}$, $\beta = 1/4$,

Table 4: Empirical false-alarm rates for independent Bernoulli outputs on the null boundary. Each setting uses 10,000 repetitions; the largest Monte Carlo standard error is 0.0044.

		$a = 0.30$		$a = 0.50$	
n	T	0.05	0.10	0.05	0.10
200	50	0.0720	0.1262	0.0617	0.1145
600	50	0.0534	0.0980	0.0480	0.0908
1000	50	0.0473	0.0854	0.0422	0.0803
1400	50	0.0401	0.0765	0.0401	0.0754
200	100	0.1104	0.1840	0.0840	0.1461
600	100	0.0697	0.1287	0.0595	0.1080
1000	100	0.0599	0.1140	0.0551	0.0989
1400	100	0.0582	0.1092	0.0468	0.0899
200	150	0.1390	0.2237	0.1056	0.1760
600	150	0.0828	0.1498	0.0644	0.1261
1000	150	0.0690	0.1305	0.0591	0.1117
1400	150	0.0650	0.1217	0.0615	0.1098
200	200	0.1625	0.2564	0.1267	0.2096
600	200	0.0964	0.1721	0.0792	0.1411
1000	200	0.0759	0.1367	0.0688	0.1243
1400	200	0.0702	0.1266	0.0649	0.1179

and 10,000 independent repetitions per setting, with the plug-in variance estimator. Table 4 reports the frequency of at least one rejection by time T, using critical values 2.52917 and 2.28337 at nominal levels 0.05 and 0.10, respectively. Calibration improves with increasing n, and the rejection rates are close to or below nominal levels for $T = 50$ and $n \geq 600$. Longer horizons exhibit size inflation, particularly for small n.

B Background on statistics and probability

B.1 Background on sequential monitoring

We provide a minimal background on the statistical literature on sequential monitoring, that is related to our present task of monitoring for harmful changes in differential privacy (DP) while controlling the false alarm rate (FAR), as detailed in Section 3. The paradigm of monitoring with FAR control was introduced in the seminal work of [31] and has since become a cornerstone of statistical real-time data analysis. For a recent overview, see [21]. Alternative approaches to real-time analysis include methods developed in the field of "Stochastic Process Control." For an overview, we refer to the monograph of [32]. Statistical monitoring has been developed to various data types, including real-valued data [33], panel data [34], and more general data objects [35]. The challenge of achieving short detection delays, which is also a focus of this paper, has been addressed by [36], [37], and [22].

To determine the relation of this body of literature to our own work, notice that both fundamentally rely on stochastic process theory to control the FAR. However, our analysis critically differs in its objective. Classical statistical monitoring typically aims to detect distributional changes in a data stream compared to a baseline period. For such methods, any distributional change is relevant. In contrast, *our approach is tailored to applications in DP, where algorithmic modifications are of interest only if they are harmful* (i.e., introduce privacy breaches). Another difference is that our framework does not rely on a "baseline period" to draw comparisons. Rather, we monitor adherence to a fixed ε-DP threshold, ensuring that it is not violated.

B.2 The standard Brownian motion

A stochastic process S indexed in an interval I is a random function $S : I \to \mathbb{R}$. This means that for any fixed value $u \in I$ the value $S(u)$ is a random variable. If S is continuous (for any random outcome), S can be characterized using its marginal distributions, i.e., distributions of the vector

$(S(u_1), \cdots, S(u_R))$ for collections of points $u_1, \cdots, u_R \in I$. If these vectors follow Gaussian distributions, then S is called a Gaussian process.

The single most important Gaussian process in applied mathematics is the standard Brownian motion (sBm), which is indexed in $I = [0, 1]$. At each point u, $B(u)$ follows a normal distribution $\mathcal{N}(0, u)$, meaning it has mean zero and variance u. The behavior of the sBm is fully characterized by its covariance structure for $u, v \in [0, 1]$:

$$\mathbb{C}ov(B(u), B(v)) = \min(u, v).$$

For a formal introduction to the sBm and its key properties, we refer to [38]. The sBm was first used in physics to describe the random motion of particles in a fluid, a phenomenon now called "Brownian motion". Since then, it has become fundamental in areas such as economics, biology, and quality control. Mathematically, the sBm can be viewed as a standard Gaussian distribution that outputs random functions, rather than just random numbers. Just as the central limit theorem describes the convergence of sums of random numbers to a Gaussian distribution, there is a version for random functions where the limit is the sBm (see, e.g., [39]). To develop some intuition, we show two realizations of the sBm (generated with different random seeds) in Figure 8 (left).

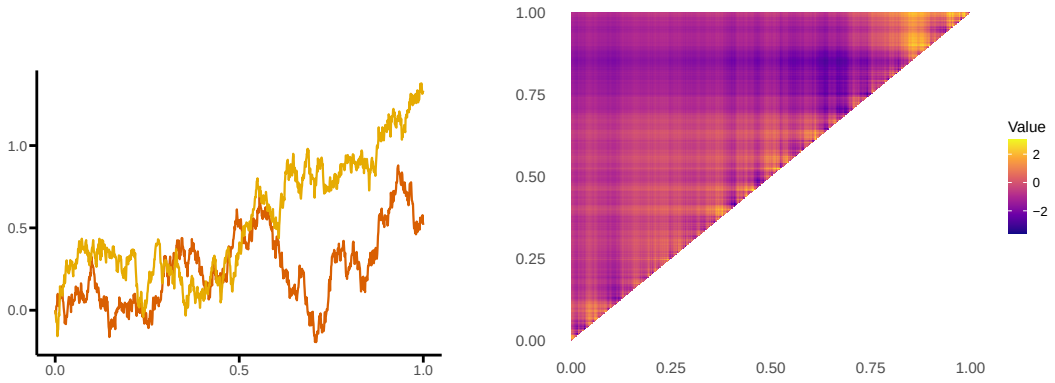


Figure 8: Left: Two realizations of the standard Brownian motion. Right: A sample realization of the function $d(u, v)$ defined in (13) over the domain $0 \leq v < u \leq 1$. This illustrates the stochastic behavior of the scaled Brownian increments used in threshold calibration.

B.3 Stochastic processes on Hölder spaces

The mathematics of this paper are rooted in stochastic process theory, a branch of probability theory. More precisely, we study stochastic processes that are Hölder continuous. We now give a rough outline of the involved objects and refer for a more detailed introduction to the work of [40].
We consider a continuous function $f : [0, 1] \to \mathbb{R}$. For a parameter $\beta \in [0, 1/2)$ we define its Hölder norm

$$\|f\|_\beta := \sup_{t \in [0,1]} |f(t)| + \sup_{0 \leq s < t \leq 1} \frac{|f(t) - f(s)|}{|t - s|^\beta}.$$

Moreover, we say that f has a vanishing Hölder modulus of continuity if

$$\lim_{\delta \downarrow 0} \sup_{\substack{0 \leq s < t \leq 1 \\ |s-t| \leq \delta}} \frac{|f(t) - f(s)|}{|t - s|^\beta} = 0.$$

The collection of all continuous functions f that have finite β-Hölder norm $\|f\|_\beta < \infty$ and have vanishing Hölder modulus of continuity make up the Hölder space $\mathcal{H}_\beta[0, 1]$. Sometimes we will define the subspace $\mathcal{H}_\beta[0, \theta]$ which consists of the restrictions on that interval (the Hölder norm is defined in analogy). The Hölder space is for any $\beta \in [0, 1/2)$ a separable Banach space and for $\beta = 0$ it is identical to the continuous functions.
A stochastic process on $\mathcal{H}_\beta[0, 1]$ (also: with sample paths in $\mathcal{H}_\beta[0, 1]$) is a measurable random variable that takes values in $\mathcal{H}_\beta[0, 1]$. One example is the sBm. For a sequence of stochastic

processes $(P_n)_n$ in $\mathcal{H}_\beta[0,1]$ to converge to a limit P in $\mathcal{H}_\beta[0,1]$, this means that P_n converges weakly w.r.t. the topology on $\mathcal{H}_\beta[0,1]$ created by the Hölder norm. To understand in detail what this means we refer to Section 2 in [40], where also additional references are given.

B.4 Simulation of the threshold $q(\alpha)$

The threshold $q(\alpha)$ used in Algorithm 1 is the upper α-quantile of the random variable

$$D = \sup_{0 \le v < u \le 1} \frac{B(u) - B(v)}{(u-v)^\beta},$$

where B denotes a standard Brownian motion and $\beta \in [0, 1/2)$. Since the distribution of D is not available in closed form, we approximate $q(\alpha)$ by Monte Carlo simulation. For this purpose, we discretize the unit interval by the grid $0 = s_0 < s_1 < \cdots < s_M = 1$, where $s_j = j/M$. We simulate Brownian motion on this grid by setting $B(s_0) = 0$ and

$$B(s_j) - B(s_{j-1}) \sim \mathcal{N}(0, 1/M), \qquad j = 1, \dots, M,$$

independently. For each simulated Brownian path, we compute the discrete approximation

$$D_M = \max_{0 \le i < j \le M} \frac{B(s_j) - B(s_i)}{(s_j - s_i)^\beta}, \quad \text{where} \quad B(s_j) - B(s_i) = \sum_{k=i+1}^{j} \big(B(s_k) - B(s_{k-1})\big).$$

Repeating this procedure R times gives independent realizations $D_M^{(1)}, \dots, D_M^{(R)}$. We then approximate $q(\alpha)$ by the empirical $(1-\alpha)$-quantile of these values and a summary is given in Algorithm 3 below. In practice we recommend using the values $R = 5000$ and $M = 1000$. The value α depends on the acceptable FAR and therefore fundamentally has to be determined by the user. However, default values suggested by standard statistical practice suggest using $\alpha = 0.05$ or $\alpha = 0.01$

Algorithm 3 Monte Carlo approximation of $q(\alpha)$

Input: Error level α, weight parameter β, grid size M, number of simulations R
Output: Approximation $\hat{q}(\alpha)$ of the upper α-quantile of D

1: **for** $r = 1, \dots, R$ **do**
2: Set $B_0^{(r)} = 0$
3: **for** $j = 1, \dots, M$ **do**
4: Draw $\xi_j^{(r)} \sim \mathcal{N}(0, 1/M)$ independently
5: Set $B_j^{(r)} = B_{j-1}^{(r)} + \xi_j^{(r)}$
6: Compute

$$D_M^{(r)} = \max_{0 \le i < j \le M} \frac{B_j^{(r)} - B_i^{(r)}}{((j-i)/M)^\beta}.$$

7: Set $\hat{q}(\alpha)$ equal to the empirical $(1-\alpha)$-quantile of $D_M^{(1)}, \dots, D_M^{(R)}$
8: **return** $\hat{q}(\alpha)$

C Proofs for our main results

We now turn to the proofs of our theoretical results. Throughout the proofs, we will use the notation $\lesssim$ when comparing two sequences of positive numbers $(a_n)_n$ and $(b_n)_n$. More precisely, we write $a_n \lesssim b_n$ if there exists a constant $C > 0$ (independent of n) such that $a_n \le C b_n$ for all $n \ge 1$. For a sequence of random variables $(Z_n)_n$ we denote weak convergence to a limit Z by $Z_n \xrightarrow{d} Z$.

C.1 Proofs for the naive method

In this section, we prove the normal approximation in (8) for fixed output distribution at t and for $n \to \infty$. For this convergence to hold it is necessary that $\hat{p}_t$ does not have a degenerate distribution

which means that its variance is > 0 (i.e. $\sigma_t^2 > 0$ defined in (19) below). If (8) holds, or more mathematically, if the weak convergence

$$\frac{\hat{p}_t - p_t}{\hat{\sigma}_t} \xrightarrow{d} \mathcal{N}(0,1) \tag{18}$$

holds it follows directly that

$$\lim_{n\to\infty} \mathbb{P}\big(p_t \in \mathcal{C}_t(\alpha)\big) = 1 - \alpha.$$

To show (18), we first compute the variance of $\hat{p}_t$:

$$\begin{aligned}\sigma_t^2 :=& n\mathbb{V}ar(\hat{p}_t) = \frac{\mathbb{V}ar(n_t^X) + e^{2\varepsilon}\,\mathbb{V}ar(n_t^Y)}{n} \\ =& q_t^X(1-q_t^X) + e^{2\varepsilon} q_t^Y(1-q_t^Y).\end{aligned} \tag{19}$$

Here, we have defined the success probabilities of the binomials involved as

$$q_t^X := \mathbb{P}(A(x) \in \mathcal{E}), \quad q_t^Y := \mathbb{P}(A(x') \in \mathcal{E}).$$

For the statement in (18), it suffices to show that $n\hat{\sigma}_t^2$ converges to σ_t^2 in probability. We demonstrate a stronger statement that has additional use when studying the aggregated auditor below. Namely we show

$$\max_t |\sigma_t^2 - n\hat{\sigma}_t^2| = o_P(1). \tag{20}$$

Here "$o_P(1)$" means convergence to 0 stochastically. We know that there exists an even natural number M such that $T \lesssim n^{M/4}$ (Assumption 1). Then we have, using Lemma 2.2.2 in [39], that

$$\begin{aligned}\mathbb{E}\max_t |\sigma_t^2 - n\hat{\sigma}_t^2| \lesssim& T^{1/M}\{\mathbb{E}\max_t |\sigma_t^2 - n\hat{\sigma}_t^2|^M\}^{1/M} \\ \lesssim& n^{1/4}\{\mathbb{E}\max_t |\sigma_t^2 - n\hat{\sigma}_t^2|^M\}^{1/M}.\end{aligned}$$

The final step of proving

$$\{\mathbb{E}\max_t |\sigma_t^2 - n\hat{\sigma}_t^2|^M\}^{1/M} = o(n^{-1/4}) \tag{21}$$

follows by standard techniques (using uniform boundedness of moments for standardized binomial random variables) and is hence omitted.
Next, it is clear (by linearity of the expectation) that $\mathbb{E}\hat{p}_t = p_t$ and finally, using the classical central limit theorem for binomial random variables that

$$n^{1/2}(\hat{p}_t - \mathbb{E}\hat{p}_t) = n^{1/2}(\hat{p}_t - p_t) \to \mathcal{N}(0, \sigma_t^2).$$

Now, combining this with the fact that $n\hat{\sigma}_t^2 - \sigma_t^2 = o_P(1)$ and using Slutsky's theorem yields (18).

C.2 Preliminary results on stochastic processes

A key technique for our proofs is reducing the convergence of a (more complicated) stochastic process P_n, to convergence of its simpler segments. For this purpose we introduce the following "gluing operation", where a stochastic process is cut into separate segments that converge and finally put together again to show convergence of the entire process.

Gluing operations for stochastic processes Suppose we have a stochastic process (random function) $P_n(x), 0 \le x \le 1$ indexed in n. Moreover, suppose that there exists a fixed number of points $\theta_0 := 0 < \theta_1 < \cdots < \theta_K < \theta_{K+1} := 1$ such that for each $k = 0, ..., K-1$ we define

$$G_n^{(k)}(x) := P_n(x + \theta_k) - P_n(\theta_k),\ 0 \le x \le \theta_{k+1} - \theta_k.$$

Further suppose that each $G_n^{(k)}$ converges weakly to an sBm $B^{(k)}$ in the Hölder space $\mathcal{H}^\beta[0, \theta_{k+1} - \theta_k]$ and the sBms are independent across k. Then, $P_n(x), 0 \le x \le 1$ converges weakly to $B(x), 0 \le x \le 1$ in the Hölder space $\mathcal{H}^\beta[0,1]$, where B is an sBm.

Proof of gluing statement The Hölder space $\mathcal{H}^\beta[0, \theta_{k+1} - \theta_k]$ is complete and separable, and thus the same is true of the product space

$$\prod_{k=0}^{K} \mathcal{H}^\beta[0, \theta_{k+1} - \theta_k]$$

equipped with the maximum metric. Using Skorohod's theorem (see [41], Theorem 6.7), we can switch to a probability space where $G_n^{(k)}$ converges almost surely to $B^{(k)}$ in the Hölder metric. Recall that P_n (which is the glued-together version of $G_n^{(k)}$) is defined for $\theta_k \leq x \leq \theta_{k+1}$ as

$$P_n(x) = G_n^{(k)}(x - \theta_k) + \sum_{\ell=0}^{k-1} G_n^{(\ell)}(\theta_{\ell+1} - \theta_\ell).$$

This object now converges almost surely to B, which is the glued together sBm, defined in $\theta_k \leq x \leq \theta_{k+1}$ as

$$B(x) = B^{(k)}(x - \theta_k) + \sum_{\ell=0}^{k-1} B^{(\ell)}(\theta_{\ell+1} - \theta_\ell).$$

It is trivial to check that B is an sBm and we omit the proof. To see the almost sure convergence of P_n, fix the outcome on the probability space such that all $G_n^{(k)}$ converge to $B^{(k)}$. Then, we have

$$\|B - P_n\|_\beta \leq \sum_k \|B^{(k)} - G_n^{(k)}\|_\beta \to 0.$$

C.3 Proof of Theorem 1

The proof of this theorem involves a number of stochastic processes and sums that are (minor) modifications of each other. Since it is easy to lose track of what a specific modification means and why it is made, we have added an overview in the below Infobox near the end of this section.

Preliminary considerations In this proof we denote $A^{(1)} := A_1$, $A^{(2)} := A_{t_1}, \cdots, A^{(K+1)} := A_{t_K}$ (to refer to the algorithm in any of the periods where it does not undergo a change. We also define the objects

$$q^{(k),X} := \mathbb{P}(A^{(k)}(x) \in \mathcal{E}), \qquad q^{(k),Y} := \mathbb{P}(A^{(k)}(x') \in \mathcal{E}).$$

Now, we will assume in this proof that for $k = 1, ..., K+1$ it holds that

$$1 > c_1 > q^{(k),X}, q^{(k),Y} > c_0 > 0$$

The (easy) modification where this value is e.g. $= 0$ for some k is ignored for the sake of compactness in this and the following proofs. We will furthermore only consider the case in this proof where $T \to \infty$ as $n \to \infty$. The case where T is fixed and $n \to \infty$ is again much easier.

Gluing and weak convergence For the proof of the theorem, we consider the stochastic process

$$P_n(x) := \sum_{t=1}^{\lfloor xT \rfloor} \frac{\hat{p}_t - p_t}{\hat{\sigma}_t \sqrt{T}} + \big[Tx - \lfloor Tx \rfloor\big] \frac{\hat{p}_{\lceil Tx \rceil} - p_{\lceil Tx \rceil}}{\hat{\sigma}_{\lceil Tx \rceil} \sqrt{T}}.$$

This process depends on n by the definition of $\hat{p}_t, \hat{\sigma}_t$ and T. The main step of this proof is now showing weak convergence of P_n to an sBm on the Hölder space $\mathcal{H}_\beta[0,1]$, i.e.

$$\{P_n(x) : 0 \leq x \leq 1\} \xrightarrow{d} \{B(x) : 0 \leq x \leq 1\}.$$

According to our considerations in Section C.2, it suffices to show separately that the processes $G_n^{(k)}$ (defined there) converge individually to an sBm on the subintervals $[0, \theta_{k+1} - \theta_k]$. Each subinterval can be treated in the same way and hence, we focus on the one with $k = 0$, on which $G_n^{(0)} = P_n$. Thus, we now need to show

$$\{P_n(x) : 0 \leq x \leq \theta_1\} \xrightarrow{d} \{B(x) : 0 \leq x \leq \theta_1\},$$

to obtain convergence.[1] From now on, we will consider P_n and all other following processes restricted to the Hölder space $\mathcal{H}^\beta[0, \theta_1]$. We define

$$\begin{aligned}\widetilde{P}_n(x) =& P_n(x)\mathbb{I}\{0 \leq x \leq \lfloor T\theta_1 \rfloor / T\} \\ &+ P_n(\lfloor T\theta_1 \rfloor / T)\mathbb{I}\{x > \lfloor T\theta_1 \rfloor / T\}\end{aligned}$$

[1] A technicality that we do not discuss in detail is the independence of the limits $B^{(k)}$ of $G_n^{(k)}$ for $k = 0, ..., K-1$, required for the gluing result in Section C.2. We just summarize the main idea here: The processes $G_n^{(k)}$ are already independent when $(T\theta_k)_k$ are all natural numbers. In general the processes $G_n^{(k)}$ are "almost independent", in the sense that they are independent except for a negligible error of size $\mathcal{O}_P(T^{-1/2})$ that asymptotically vanishes. Thus their limits are completely independent.

and elementary calculations show that

$$\|\widetilde{P}_n - P_n\|_\beta = o_P(1).$$

What $\widetilde{P}_n$ does is that it only uses data from before the modification time θ_1 (the very short period $\lfloor T\theta_1 \rfloor / T < x \leq T\theta_1$ it then remains constant). Given the asymptotic identity of $P_n, \widetilde{P}_n$, it suffices to show the weak convergence for $\widetilde{P}_n$ to an sBm B.

Weak convergence of $\widetilde{P}_n$ At the beginning of this proof, we said that $\hat{p}_t$ has a positive variance for any t (see eq. (19)). Recall that the true variance of $\hat{p}_t$ is σ_t^2/n. Since $\widetilde{P}_n$ only involves information of time points $1 \leq t \leq t_1 = \lfloor T\theta_1 \rfloor$, all of the $\hat{p}_t$ are i.i.d. and have the same positive variance, which is σ_1^2/n. Using the finding in (20), it follows that with probability going to 1 it holds that (for all t simultaneously)

$$\frac{\hat{p}_t - p_t}{\hat{\sigma}_t} = \frac{\hat{p}_t - p_t}{\hat{\sigma}_t} \mathbb{I}\{n^{1/2}\hat{\sigma}_t \geq \sigma_1/2\}.$$

Defining for $t = 1, ..., \lfloor T\theta_1 \rfloor$

$$Z_t := \frac{\hat{p}_t - p_t}{\hat{\sigma}_t} \mathbb{I}\{n^{1/2}\hat{\sigma}_t \geq \sigma_1/2\}$$

and $Z_{\lfloor T\theta_1 \rfloor + 1}$ as an independent copy of these (i.i.d.) variables, we obtain that

$$\widetilde{P}_n(x) = P_n'(x) + o_P(1),$$

where the remainder is vanishing in the Hölder norm. Here

$$P_n'(x) := \sum_{i=1}^{\lfloor xT \rfloor} \frac{Z_i}{\sqrt{T}} + \big[Tx - \lfloor Tx \rfloor\big] \frac{Z_{\lceil Tx \rceil}}{\sqrt{T}}.$$

Now, we may show weak convergence of P_n' on the Hölder space to an sBm. According to Section 2.1 in [40] (observations a) and b) on p. 265) we need to show weak convergence of the marginal distributions of P_n' and tightness in the Hölder space. If we can show marginal convergence, tightness already follows from criterion ii) on p. 266 in [40]. This criterion can be validated following exactly the steps in the proof of Lemma 3.3 in [22]. This leaves us with proving marginal convergence. In principle the proof needs to be conducted using multivariate marginals, but for illustration we will confine ourselves to showing that $P_n'(\theta_1)$ converges to a standard normal.
We have that

$$P_n'(\theta_1) = \sum_{i=1}^{\lfloor T\theta_1 \rfloor} \frac{Z_i}{\sqrt{T}} + o_P(1) =: C_1 + o_P(1).$$

Here C_1 is defined in the obvious way. Next, we introduce the variables

$$\check{Z}_t := \frac{\hat{p}_t - p_t}{\sigma_t / n^{1/2}} \mathbb{I}\{n^{1/2}\hat{\sigma}_t \geq \sigma_1/2\}$$

and therewith

$$C_2 := \sum_{i=1}^{\lfloor T\theta_1 \rfloor} \frac{\check{Z}_i}{\sqrt{T}}.$$

We now demonstrate that $C_1 - C_2 = o_P(1)$. By Cauchy–Schwarz, using $\lfloor T\theta_1 \rfloor \leq T$, we obtain

$$\begin{aligned}
&\mathbb{E}\big[(C_1 - C_2)^2\big] \\
&\leq \sum_t \mathbb{E}\bigg[\Big(\frac{(\hat{p}_t - p_t)(\hat{\sigma}_t - \sigma_t/n^{1/2})}{(\sigma_t/n^{1/2})\hat{\sigma}_t}\Big)^2 \mathbb{I}\{n^{1/2}\hat{\sigma}_t \geq \sigma_1/2\}\bigg] \\
&= \sum_t \mathbb{E}\bigg[\Big(\frac{n^{1/2}(\hat{p}_t - p_t)(n^{1/2}\hat{\sigma}_t - \sigma_t)}{\sigma_t(n^{1/2}\hat{\sigma}_t)}\Big)^2 \mathbb{I}\{n^{1/2}\hat{\sigma}_t \geq \sigma_1/2\}\bigg] \\
&\lesssim T\,\mathbb{E}\big[\{n^{1/2}(\hat{p}_t - p_t)(n^{1/2}\hat{\sigma}_t - \sigma_t)\}^2\big] \\
&\leq T\,\{\mathbb{E}(n^{1/2}[\hat{p}_t - p_t])^4\}^{1/2}\{\mathbb{E}(n^{1/2}\hat{\sigma}_t - \sigma_t)^4\}^{1/2}.
\end{aligned}$$

It is elementary to show that

$$\{\mathbb{E}(n^{1/2}[\hat{p}_t - p_t])^4\}^{1/2} = \mathcal{O}(1).$$

Moreover, since $\sigma_t = \sigma_1 > 0$ on the segment under consideration, the fourth-moment bounds for binomial proportions yield

$$\{\mathbb{E}(n^{1/2}\hat{\sigma}_t - \sigma_t)^4\}^{1/2} \leq \frac{1}{\sigma_t^2}\{\mathbb{E}(n\hat{\sigma}_t^2 - \sigma_t^2)^4\}^{1/2} = \mathcal{O}(n^{-1}).$$

Thus, $\mathbb{E}[(C_1 - C_2)^2] = \mathcal{O}(T/n) = o(1)$ under $T = o(n)$, and the claim that $C_1 - C_2 = o_P(1)$ follows directly. Next, we define

$$C_3 := \sum_{i=1}^{\lfloor T\theta_1 \rfloor} \frac{\bar{Z}_i}{\sqrt{T}},$$

where

$$\bar{Z}_t := \frac{\hat{p}_t - p_t}{\sigma_t / n^{1/2}}.$$

Using again the fact that $n^{1/2}\hat{\sigma}_t \geq \sigma_1/2$ holds asymptotically with probability going to 1 and simultaneously for all t, it follows that $C_2 = C_3$ with probability going to 1. Now, C_3 is easy to investigate. Asymptotic normality for C_3 can now be established using the Lindeberg central limit theorem. Notice that the variance of $\bar{Z}_t/\sqrt{T}$ is by construction $1/T$. It follows that

$$C_3 \xrightarrow{d} \mathcal{N}(0, \theta_1).$$

This finishes the proof of the fact that P_n converges to an sBm on the Hölder space. The next lemma shows weak convergence of a version of $\widehat{D}$.

Summary of stochastic processes and their use in the proofs

1) $(B^{(k)})$, B: (segment of) sBm; it occurs as limiting distribution on the Hölder space $\mathcal{H}_\beta$ of P_n.
2) P_n: partial sum process of the statistics $(\hat{p}_t - p_t)/\hat{\sigma}_t$; it is possible to write the detector $\widehat{D}(\tau)$ using P_n and thus studying the convergence of P_n is integral to our proof.
3) $\widetilde{P}_n$: We study the convergence of P_n in the proof of Theorem 1 restricted to the interval $[0, \theta_1]$; but $\lfloor T\theta_1 \rfloor/T$ may be smaller than θ_1 and thus P_n uses information from after the change point θ_1; to avoid this complication we create $\widetilde{P}_n$ which only uses data from times $1 \leq t \leq \lfloor T\theta_1 \rfloor$.
4) P_n': the individual terms in $\widetilde{P}_n$ are divided by $\hat{\sigma}_t$; but $\hat{\sigma}_t$ can be close to 0 (with small probability); these cases make it hard to study weak convergence; so we move to a process where $\hat{\sigma}_t$ (times $n^{1/2}$) is bounded away from 0, namely P_n'.
5) C_1, C_2, C_3: $C_1 = P_n'(\theta_1)$, i.e. a sum of i.i.d. random variables; to more conveniently apply a CLT we replace C_1 by C_3 which consists of i.i.d. variables that all have variance $1/T$; C_2 is used for an intermediary replacement step.

Lemma 1. *Suppose that on the Hölder space $\mathcal{H}^\beta[0,1]$ the weak convergence*

$$\{P_n(x) : 0 \leq x \leq 1\} \xrightarrow{d} \{B(x) : 0 \leq x \leq 1\}.$$

holds, where B is an sBm. Define

$$\widetilde{D}(\tau) := \max_{0 \leq \ell < \tau} \left[\frac{1}{(\ell+1)^\beta \cdot T^{1/2-\beta}} \sum_{t=\tau-\ell}^{\tau} \frac{\hat{p}_t - p_t}{\hat{\sigma}_t} \right].$$

Then, it also follows that

$$\max_{1 \leq \tau \leq T} \widetilde{D}(\tau) \xrightarrow{d} \sup_{0 \leq x < y \leq 1} \frac{B(y) - B(x)}{(y-x)^\beta} = D.$$

For the proof, the reader first needs to observe that by some additional arguments

$$\max_{1\leq\tau\leq T}\widetilde{D}(\tau)=\sup_{0\leq x<y\leq 1}\frac{P_n(y)-P_n(x)}{(y-x)^\beta}+o_P(1).$$

Now, the lemma is a direct consequence of the continuous mapping theorem, combined with the fact that the mapping

$$\mathcal{H}^\beta[0,1]\ni f\mapsto\sup_{0\leq x<y\leq 1}\frac{f(y)-f(x)}{(y-x)^\beta}\in\mathbb{R}$$

is continuous. Because then we have

$$\sup_{0\leq x<y\leq 1}\frac{P_n(y)-P_n(x)}{(y-x)^\beta}\xrightarrow{d}\sup_{0\leq x<y\leq 1}\frac{B(y)-B(x)}{(y-x)^\beta}.$$

Last step To finish the proof of the theorem, we need to consider the two cases, where either $\max_t p_t\leq 0$ (NPVT) or $\max_t p_t>0$ (not NPVT). We focus here on the NPVT because the other case is discussed in the proof of Theorem 2. If NPVT holds, we get for all τ

$$\widehat{D}(\tau)\leq\max_{0\leq\ell<\tau}\left[\frac{1}{(\ell+1)^\beta\cdot T^{1/2-\beta}}\sum_{t=\tau-\ell}^{\tau}\frac{\hat{p}_t-p_t}{\hat{\sigma}_t}\right].$$

For the right side (which is $\widetilde{D}(\tau)$), we have shown above weak convergence to D (defined in (13)). Consequently

$$\begin{aligned}&\mathbb{P}\big(O=\text{"DP violation detected"}\big)\\=&\mathbb{P}\big(\max_\tau\widehat{D}(\tau)>q(\alpha)\big)\leq\mathbb{P}\big(\max_\tau\widetilde{D}(\tau)>q(\alpha)\big)\to\mathbb{P}\big(D>q(\alpha)\big)=\alpha.\end{aligned}$$

This completes the proof.

C.4 Proof of Corollary 1

The corollary follows by an investigation of the proof of Theorem 1. More precisely, since $c_n\to 0$, it follows for all sufficiently large n that $c_n<\sigma_1/2$. This means that in the definition of Z_t it does not matter whether we have $\hat{\sigma}_t$ or the stabilized variance estimator $\tilde{\sigma}_t(c_n)$. The rest of the proof then proceeds without any changes.

C.5 Details on delay times

Part ii) of Theorem 1 implies that (asymptotically) any violation of the NPVT condition will be detected by our procedure. A lingering question is how long it takes after a "bad modification" to detect it. More precisely, let us suppose that NPVT is violated and the violation occurs first after the modification t_{k^*} for some $k^*\in\{1,...,K\}$. We denote by τ^* the first time $\geq t_{k^*}$, where Algorithm 1 gives out "DP violation detected". A relevant question for the user is how large the difference $\tau^*-t_{k^*}$ is, i.e. how long after a bug is introduced it takes to detect it. The gap between signal and detection is canonically called "detection delay" in the literature [21] and shorter delays are preferable. In the case of DP monitoring a shorter delay means that an algorithmic bug is faster exposed and thus the deployment of a faulty algorithm can be stopped faster. For the statement of the next result, we need the stochastic Landau symbol denoted by "$\mathcal{O}_P$". Basically for a sequence of positive numbers $(a_n)_n$ the symbol $\mathcal{O}_P(a_n)$ refers to a product of $a_n\cdot X$ where X is a random variable (i.e. we have a rate of a_n times a random factor that does not depend on n).

Theorem 2 states that delay times are very short (in particular much shorter than any fixed fraction of T). Larger values of β imply shorter detection delays, at the cost of slightly worse approximations for the level α. Larger values of n imply shorter detection delay, which makes sense because they mean larger samples and hence more precise estimates.

Let us now outline the proof: Without loss of generality we assume that the privacy violation we are searching happens already during the first observation period and thus $p_1>0$. Setting $\ell=\tau-1$ we

get at time $\tau < \lfloor T\theta_1 \rfloor$ that

$$\begin{aligned}\widehat{D}(\tau) \geq & \left[\frac{1}{(\tau+1)^{\beta} \cdot T^{1/2-\beta}} \sum_{t=1}^{\tau} \frac{\hat{p}_t - p_t}{\hat{\sigma}_t}\right] \\ & + \left[\frac{1}{(\tau+1)^{\beta} \cdot T^{1/2-\beta}} \sum_{t=1}^{\tau} \frac{p_t}{\hat{\sigma}_t}\right] \\ =: & R_1 + R_2.\end{aligned}$$

where $\widehat{D}(\tau)$ is defined in (11). Obviously $R_1 \leq \widetilde{D}(\tau)$, and we know that (uniformly in τ) we have

$$R_1 \leq \max_{\tau} \widetilde{D}(\tau) = \mathcal{O}_P(1)$$

due to Lemma 1. Thus, we study R_2. We know that $\sigma_1 > 0$. From (21) we have with probability going to 1 the inequality that

$$R_2 \geq \left[\frac{1}{(\tau+1)^{\beta} \cdot T^{1/2-\beta}} \sum_{t=1}^{\tau} \frac{p_t}{\sigma_1/2}\right] \geq c_0 \frac{\tau^{1-\beta}\sqrt{n}}{T^{1/2-\beta}},$$

where c_0 is a sufficiently small, positive constant. The right side can be made arbitrary large if, for some sufficiently large constant c_τ, we choose

$$\tau := c_\tau \left\lfloor \frac{T^{(1/2-\beta)/(1-\beta)}}{n^{1/(2(1-\beta))}} \right\rfloor$$

and in particular delay times are of smaller (or equal) rate as the right side. To understand this conclusion, notice that R_1 is stochastically bounded, but we can make R_2 arbitrarily large and thus, we can make $R_1 + R_2$ arbitrarily large, in particular larger than $q(\alpha)$, which then leads to detection.

D Temporally dependent observations

We extend false-alarm control to a setting with temporal dependence. More precisely, we will assume temporal dependence (along t), but cross-sectional independence (along i). Fix $(x, x', \mathcal{E})$ and ε, and write

$$Z_{i,t} := \mathbf{1}\{X_{i,t} \in \mathcal{E}\} - e^{\varepsilon}\mathbf{1}\{Y_{i,t} \in \mathcal{E}\}, \qquad p_t := \mathbb{E}Z_{i,t}, \qquad \hat{p}_t := \frac{1}{n}\sum_{i=1}^{n} Z_{i,t}. \tag{22}$$

As in Section 3.2, let $t_k = \lfloor T\theta_k \rfloor$, $0 < \theta_1 < \cdots < \theta_K < 1$. Set $t_0 = 1$, $t_{K+1} = T + 1$, $I_k = \{t_k, \ldots, t_{k+1} - 1\}$ and $k(t) = k$ for $t \in I_k$. The asymptotic index n is suppressed in the observations.

Definition 2 (Strong mixing)**.** For two σ-fields $\mathcal{A}, \mathcal{B}$, define

$$\alpha(\mathcal{A}, \mathcal{B}) := \sup_{A \in \mathcal{A},\, B \in \mathcal{B}} |\mathbb{P}(A \cap B) - \mathbb{P}(A)\mathbb{P}(B)|.$$

The contrast array is uniformly strongly mixing if $a(h) \to 0$, where

$$a(h) := \sup_{n : T(n) > h} \max_{1 \leq r \leq T(n) - h} \alpha\left(\sigma(Z_{1,t} : 1 \leq t \leq r), \sigma(Z_{1,t} : r + h \leq t \leq T(n))\right), \quad h \geq 1. \tag{23}$$

Assumption 2 (Temporal dependence)**.** For constants $0 < \rho < 1$ and $c_0 \in (0, \infty)$,

$$n \to \infty, \qquad T = T(n), \qquad T/n^{\rho} \to c_0.$$

The number K and the proportions $(\theta_k)_{k=1}^{K}$ are fixed. The trajectories $((X_{i,t}, Y_{i,t}))_{t=1}^{T}$, $i = 1, \ldots, n$, are independent and identically distributed. For each k, there is a fixed strictly stationary sequence $(Z_t^{(k)})_{t \in \mathbb{Z}}$ such that

$$\mathcal{L}((Z_{i,t})_{t \in I_k}) = \mathcal{L}((Z_t^{(k)})_{t \in I_k}), \qquad p^{(k)} := \mathbb{E}Z_0^{(k)}.$$

For constants $C, c > 0$, $a(h) \leq Ce^{-ch}$ for every $h \geq 1$. With $\gamma_k(h) := \mathrm{Cov}(Z_0^{(k)}, Z_h^{(k)})$, assume

$$s_k^2 := \gamma_k(0) + 2\sum_{h=1}^{\infty} \gamma_k(h) > 0, \qquad k = 0, \ldots, K. \tag{24}$$

Independence is imposed across replication paths, but may linger within any single path. Moreover, dependence may persist through modification times. Both marginal distributions and temporal covariances may change between regimes. We assumed weak dependence in the sense of exponentially shrinking coefficients $a(h)$ and this ensures absolute convergence in (24).

Online normalization. Define the block-length $b = b_T := \lceil \log(eT) \rceil$ for variance estimation and the boundary parameter $c_T := 1/\log(eT)$. Monitoring starts at $t = b$, when the first complete block is available. For $t = b, \dots, T$, define

$$S_{i,t} := \sum_{u=t-b+1}^{t} Z_{i,u}, \qquad \bar{S}_t := \frac{1}{n}\sum_{i=1}^{n} S_{i,t},$$
$$\hat{s}_t^2 := \frac{1}{b(n-1)}\sum_{i=1}^{n}(S_{i,t} - \bar{S}_t)^2, \qquad \tilde{s}_t := \max(\hat{s}_t, c_T). \tag{25}$$

This is the sample variance of the block sums across replications, divided by b. It is nonnegative and uses only observations available at time t. Within regime k,

$$v_{k,b}^2 := \frac{1}{b}\operatorname{Var}\left(\sum_{u=1}^{b} Z_u^{(k)}\right) = \gamma_k(0) + 2\sum_{h=1}^{b-1}\left(1 - \frac{h}{b}\right)\gamma_k(h) = s_k^2 + \mathcal{O}(b^{-1}). \tag{26}$$

For a fixed $\beta \in [0, 1/2)$, replace the marginal standard error in the original detector by $\tilde{s}_t/\sqrt{n}$:

$$\widehat{D}_{\mathrm{dep}}(\tau) := \max_{b \le r \le \tau} \frac{\sum_{t=r}^{\tau} \sqrt{n}\,\hat{p}_t/\tilde{s}_t}{(\tau - r + 1)^{\beta} T^{1/2-\beta}}, \qquad \tau = b, \dots, T. \tag{27}$$

Theorem 3 (False-alarm control)**.** Suppose that Assumption 2 holds and $\beta \in [0, 1/2)$. Under NPVT,

$$\limsup_{n\to\infty} \mathbb{P}\left(\max_{b \le \tau \le T} \widehat{D}_{\mathrm{dep}}(\tau) > q\right) \le \alpha. \tag{28}$$

If $p^{(k)} = 0$ for every k, then $\max_{b \le \tau \le T} \widehat{D}_{\mathrm{dep}}(\tau) \Rightarrow D$, and the probability in (28) converges to α.

Proof. For illustration, we restrict the proof to the case $K = 0$. Write

$$U_t := \sqrt{n}(\hat{p}_t - p^{(0)}).$$

1. Gaussian limit. Let $P_{i,T}$ be the linearly interpolated partial sum process, defined in grid-points as

$$P_{i,T}(j/T) = \frac{1}{s_0\sqrt{T}}\sum_{t=1}^{j}(Z_{i,t} - p^{(0)}), \qquad j = 0, \dots, T.$$

The scalar, fully observed case of the coupling result in [42] gives independent pairs $(P_{i,T}, B_i)$, where the B_i are standard Brownian motions and

$$\mathbb{E}\|P_{1,T} - B_1\|_{\infty}^2 = o(1).$$

For each $u \in [0, 1]$, independence and centeredness imply

$$\mathbb{E}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\{P_{i,T}(u) - B_i(u)\}\right|^2 = \mathbb{E}|P_{1,T}(u) - B_1(u)|^2 = o(1), \qquad \frac{1}{\sqrt{n}}\sum_{i=1}^{n} B_i \overset{d}{=} B.$$

This yields finite-dimensional convergence. Boundedness, exponential mixing [43], and independence across replications give

$$\mathbb{E}\left|\sum_{t=r}^{\tau} U_t\right|^{2m} \le C_m(\tau - r + 1)^m, \qquad 1 \le r \le \tau \le T, \quad m \ge 2. \tag{29}$$

Choose m with $\beta < 1/2 - 1/(2m)$. The moment bound gives tightness of the interpolated partial sums in the Hölder norm of order β. Consequently, since $b/T \to 0$,

$$\max_{b \le r \le \tau \le T} \frac{\sum_{t=r}^{\tau} U_t/s_0}{(\tau - r + 1)^{\beta} T^{1/2-\beta}} \Rightarrow D. \tag{30}$$

2. Uniform variance estimation. Stationarity and (26) yield

$$\frac{1}{b}\operatorname{Var}(S_{1,t}) = v_{0,b}^2 = s_0^2 + \mathcal{O}(b^{-1}), \qquad t = b, \dots, T.$$

Hoeffding's inequality across replications and a union bound over t give

$$\max_{b \le t \le T} |\hat{s}_t^2 - v_{0,b}^2| = \mathcal{O}_P\left(b\sqrt{\frac{\log(eT)}{n}}\right). \tag{31}$$

Since $s_0 > 0$ and $c_T \to 0$,

$$\max_{b \le t \le T} |\tilde{s}_t^{-1} - v_{0,b}^{-1}| = \mathcal{O}_P\left(b\sqrt{\frac{\log(eT)}{n}}\right), \qquad \sum_{t=1}^{T} U_t^2 = \mathcal{O}_P(T).$$

3. Estimated normalization and null comparison. Cauchy–Schwarz gives

$$\max_{b \le r \le \tau \le T} \frac{\left|\sum_{t=r}^{\tau} U_t(\tilde{s}_t^{-1} - v_{0,b}^{-1})\right|}{(\tau - r + 1)^{\beta} T^{1/2-\beta}}$$
$$\le \max_{b \le t \le T} |\tilde{s}_t^{-1} - v_{0,b}^{-1}| \left(\sum_{t=1}^{T} U_t^2\right)^{1/2} = \mathcal{O}_P\left(\sqrt{\frac{T\log^3(eT)}{n}}\right) = o_P(1).$$

Together with $s_0/v_{0,b} \to 1$, this shows that replacing s_0 by $\tilde{s}_t$ in (30) preserves the limit. Under NPVT,

$$\frac{\sqrt{n}\,\hat{p}_t}{\tilde{s}_t} = \frac{U_t}{\tilde{s}_t} + \frac{\sqrt{n}\,p^{(0)}}{\tilde{s}_t} \le \frac{U_t}{\tilde{s}_t},$$

with equality if $p^{(0)} = 0$. Both assertions follow. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are backed by proofs, theory and experiments in the main body and appendix of our paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a dedicated discussion of limitations at the end of our paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We give detailed sets of assumptions and mathematical proofs as well as adequate background in the main paper and appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a detailed description of the algorithms used and the choice of input parameters to reproduce our experimental findings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide scripts for our monitoring procedure as supplements to our submission.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We specify all parameters, databases, sample sizes etc. to understand and reproduce our results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper provides summarizing statistics (in particular cumulative counts), averaged across many repetitions of the experiments for statistical significance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on the compute resources in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our paper aims at improving the validation and development of privacy mechanisms. We do not foresee any societal harm by this work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release data or models that are at risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: This paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The original methodology, theory, and main code were developed without LLM assistance, apart from spell-checking. During the revision, LLMs were used to check and refine mathematical arguments and code.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.