Privacy Audit as Bits Transmission: (Im)possibilities for Audit by One Run

Zihang Xiang
KAUST
zihang.xiang@kaust.edu.sa

Tianhao Wang
University of Virginia
tianhao@virginia.edu

Di Wang
KAUST
di.wang@kaust.edu.sa

Abstract

Auditing algorithms' privacy typically involves simulating a game-based protocol that guesses which of two adjacent datasets was the original input. Traditional approaches require thousands of such simulations, leading to significant computational overhead. Recent methods propose single-run auditing of the target algorithm to address this, substantially reducing computational cost. However, these methods' general applicability and tightness in producing empirical privacy guarantees remain uncertain.

This work studies such problems in detail. Our contributions are twofold: First, we introduce a unifying framework for privacy audits based on information-theoretic principles, modeling the audit as a bit transmission problem in a noisy channel. This formulation allows us to derive fundamental limits and develop an audit approach that yields tight privacy lower bounds for various DP protocols. Second, leveraging this framework, we demystify the method of *privacy audit by one run*, identifying the conditions under which single-run audits are feasible or infeasible. Our analysis provides general guidelines for conducting privacy audits and offers deeper insights into the privacy audit.

Finally, through experiments, we demonstrate that our approach produces tighter privacy lower bounds on common differentially private mechanisms while requiring significantly fewer observations. We also provide a case study illustrating that our method successfully detects privacy violations in flawed implementations of private algorithms.

1 Introduction

Safeguarding data privacy has become increasingly important in machine learning tasks, particularly more so when large language models (LLMs) are demanding more data than the whole Internet [44]. Among all privacy-enhancing technologies, differential privacy (DP) [1, 3, 15, 16, 19, 45, 46] has emerged as a leading paradigm to address these concerns with rigorous mathematical guarantees. DP ensures that the inclusion or exclusion of a single data point has a minimal impact on the outcome produced by the private algorithm.

Despite its theoretical rigor, implementing differential privacy in practice remains a significant challenge; unintended error often creeps into the realizations. For instance, the sparse vector technique (SVT), a famous differential privacy protocol, has seen erroneous (not private as claimed) applications in some work [9, 50] even though the mathematical proof is given. For another example, a refinement [38] of the DP-SGD protocol [1, 5, 11, 18, 35, 46, 47] that claims to achieve surprisingly strong performance has been proven to suffer from incorrect analysis. Errors are also seen in the implementation phase, where a random seed problem [22], or a floating-point vulnerability [28] could undermine the integrity of the privacy protocol.

For any real-world application of DP, a straightforward countermeasure is to check the proposed private protocol's analysis or to go through the implemented code line by line. However, this is often cumbersome and also susceptible to errors. These considerations motivate *privacy audit* [20, 21, 31], an empirical approach to measure the privacy provided by differentially private algorithms. It does not check/verify a targeted private algorithm in its detailed implementation; instead, it often involves simulating a *distinguishing game* where an adversary attempts to identify which of two adjacent databases was used as input to run the private algorithm. Intuitively, the targeted algorithm is suggested to be not as private as claimed once accurate identification is achieved.

One standing disadvantage is that such a distinguishing game is usually required to be repeated thousands of times, incurring thousands of times of running the targeted private algorithm itself, because the final probabilistic claim for privacy requires a substantial number of observations to reach non-trivial confidence [6, 7, 30, 36]. This makes it infeasible when running the private algorithm is expensive. Recently, Steinke et al. [37] propose an audit technique that requires the algorithm (DP-SGD) to run only once while also giving meaningful claims about the empirical privacy level of the targeted algorithm. The critical operation is to perform

membership inference [34] on multiple data examples simultaneously based on the result of one run [37] of the targeted private algorithm. In terms of efficiency, such a *privacy-audit-by-one-run* technique is a substantial improvement to previous *privacy-audit-by-multiple-run* approaches.

Motivation. The audit story does not end here. We notice several problems worthy of deeper investigation:

1. The final empirical privacy claim (known as *privacy lower bound*) by [37] is not tight in general, e.g., when auditing probably the most well-known Gaussian mechanism, [37] does not give tight results, even after parameter setups are carefully tweaked and extensively tried; it is unclear how such phenomenon happens. Is such limitation inevitable? Operating under our auditing framework, we can overcome such difficulties, i.e., we can achieve tight results even more efficiently.

2. To the appealing goal of privacy audit by one run, it is also unclear when it is possible/impossible to transfer such a method to other differentially private protocols; are there any helpful universal guidelines for implementing privacy audit by only one run of the targeted private algorithm? By our unifying language, we provide a bias-variance argument, highlighting when audit-by-one-run is possible or impossible, providing guidance on how to better leverage such a technique.

Contribution. Answering such questions requires a deeper understanding of the privacy audit itself, which is the overall goal we aim to achieve in this paper. Our contribution can be summarized in the following two main parts.

1) A unifying language for privacy audit and improved audit method. We model the privacy audit problem as bits transmission under an information-theoretic context. Behind such treatment is the observation that if some algorithm $\mathcal{M}$ is DP, it ensures indistinguishability between output due to some adjacent input dataset X, X'; determining whether it is X or X' (0 or 1) is no difference from recovering **one bit** of information. Roughly, in the distinguishing game, one chooses X or X' as input to $\mathcal{M}$, and the adversary guesses which one was chosen based on the output of $\mathcal{M}$. In analogy, it coincides with the scenario where a sender aims to communicate one bit of information to a receiver through a noisy channel, corresponding to the execution of $\mathcal{M}$.

Based on such treatment, we study the behavior of such modeling, deriving fundamental limits for bits transmission using the language of information theory. We then leverage those results to design our audit principle. At a high level, if the bits transmission can achieve very low bit error, we can claim (with confidence specification) that $\mathcal{M}$ is not private as promised.

Except for some carefully designed regulations, our modeling abstracts from the details of how auditing $\mathcal{M}$ is carried out, meaning that our framework can handle both cases of privacy-audit-by-multiple-run and privacy-audit-by-one-run. Practitioners can freely arrange their audit tasks according to regulations, and a privacy lower bound can be derived effortlessly based on our framework.

2) On the tightness of privacy audit and (im)possibilities of privacy audit by one run. Relying on our "privacy audit as bits transmission" modeling, we can answer the two previously raised questions. We show that by modeling $\mathcal{M}$ via f-DP [13] (a DP formulation based on hypothesis testing), we can get tight audit results across various DP protocols; on the other hand, interpretations for why previous work achieves loose audit results are also provided.

Then, based on our framework and theoretical analysis, we 1) demystify privacy-audit-by-one-run and reveal when only one run is (im)possible and 2) provide guidelines on avoiding sub-optimal choices or any other pitfalls for all privacy audit tasks.

In the sequel, targeting the DP-SGD protocol and with the goal of auditing by only *one run*, we carry out experiments of privacy audit on real-world tasks. We show our method achieves better lower bounds than the previous approach to the problem of *privacy audit by one run*. We also give a case study illustrating that our method successfully catches the bug in some ill-implemented private algorithms.

2 Background

2.1 Differential Privacy (DP)

Definition 1 (Differential Privacy [14])**.** *Let $\mathcal{M} : \mathcal{X}^* \to \mathcal{Y}$ be a randomized algorithm, where $\mathcal{X}^* = \bigcup_{n\geq 0} \mathcal{X}^n$. We say $\mathcal{M}$ is (ε, δ)-differentially private ((ε, δ)-DP) if, for all $X, X' \in \mathcal{X}^*$ differing only by one element, we have $\forall S \subset \mathcal{Y}$*

$$\Pr(\mathcal{M}(X) \in S) \leq e^{\varepsilon} \cdot \Pr(\mathcal{M}(X') \in S) + \delta.$$

There are two versions of adjacency: 1) for addition/removal, X' has exactly one more data sample than that of X; 2) for replacement: X and X' contain the same number of data samples but only differ in exactly one. Our framework in this paper can handle both versions. Post-processing on the output of the DP algorithm is still DP, and the execution of multiple DP algorithms sequentially, known as *composition*, also maintains DP.

A functional formulation of DP: f-DP. Using the (ε, δ)-DP to characterize the privacy of some private algorithm $\mathcal{M}$ has been shown to be lossy [13]. This is because such a single pair of parameters cannot express the rich nature of the privacy promised by $\mathcal{M}$. In contrast, f-DP, based on hypothesis testing formulation, reflects the nature of private mechanisms by a *function* f [13, 51] rather than a single pair of parameter (ε, δ).

The hypothesis testing setups for f-DP is as follows. Let $\mathcal{Y}$ be the output space of $\mathcal{M}$ taking input one dataset from

adjacent datasets X, X', we form the *null* and *alternative* hypotheses:

$$\mathbf{H_0}: X \text{ was the input}, \quad \mathbf{H_1}: X' \text{ was the input}. \tag{1}$$

For a decision rule $\mathcal{R}: \mathcal{Y} \to \{0,1\}$ for such hypothesis testing setups, two types of errors stand out:

- Type I error or false positive rate $\alpha = \Pr[\mathcal{R}(y) = 1 | \mathbf{H_0}]$, i.e., the probability of rejecting $\mathbf{H_0}$ while $\mathbf{H_0}$ is true;
- Type II error or false negative rate $\beta = \Pr[\mathcal{R}(y) = 0 | \mathbf{H_1}]$, i.e., the probability of rejecting $\mathbf{H_1}$ while $\mathbf{H_1}$ is true.

It is inevitable to make trade-offs between α and β; what is interesting is the best β one can achieve for fixed α. This is related to the following definition.

Definition 2 (Trade-off function [13]). *For a hypothesis testing problem over two distributions P, P', define the trade-off function as:*

$$T_{P,P'}(\alpha) = \inf_{\mathcal{R}} \{\beta_{\mathcal{R}} : \alpha_{\mathcal{R}} \leq \alpha\}$$

where decision rule $\mathcal{R}$ takes input a sample from P or P' and decides which distribution produced that sample. The infimum is taken over all decision rule $\mathcal{R}$.

The trade-off function quantifies the best one can do in a hypothesis-testing problem. The optimal β is achieved via the likelihood ratio test, which is also known as the fundamental *Neyman–Pearson lemma* [32] (please refer to Appendix A.1). For function f and g, we denote

$$g \geq f \text{ if } g(x) \geq f(x), \forall x \in [0,1].$$

Definition 3 (f-DP [13]). *Let $f: [0,1] \to [0,1]$ be a trade-off function. A mechanism $\mathcal{M}$ is f-DP if*

$$T_{\mathcal{M}(X),\mathcal{M}(X')} \geq f$$

holds for all adjacent dataset X, X'

f-DP formulation quantifies the indistinguishability between the output of $\mathcal{M}$ due to X or X' by a function, much more expressive than what a single pair of (ε, δ) tells. In fact, f-DP is a generalization of (ε, δ)-DP [13, 42]: $\mathcal{M}$ is (ε, δ)-DP equals to $\mathcal{M}$ is $f_{\varepsilon,\delta}$-DP where the trade-off function $f_{\varepsilon,\delta}$ is

$$f_{\varepsilon,\delta}(x) = \max(0, 1-\delta-e^{\varepsilon}x, e^{-\varepsilon}(1-\delta-x))$$

We also have a useful family of trade-off functions parameterized by μ as follows.

Definition 4 (μ-Gaussian DP (μ-GDP) [13]). *The trade-off function of distinguishing $\mathcal{N}(0,1)$ from $\mathcal{N}(\mu,1)$ is*

$$G_\mu(x) = T_{\mathcal{N}(0,1),\mathcal{N}(\mu,1)}(x) = \Phi(\Phi^{-1}(1-x) - \mu),$$

where Φ be the c.d.f. of standard normal distribution. A private mechanism $\mathcal{M}$ satisfies μ-GDP if it is G_μ-DP

2.2 Privacy Audit

An equivalent object to the trade-off function is the "testing region" for some algorithm satisfying (ε, δ)-DP by the following theorem.

Theorem 1 ((ε, δ)-DP's testing region [21]). *For any $\varepsilon > 0$ and $\delta \in [0,1]$, a mechanism $\mathcal{M}$ is (ε, δ)-DP if and only if*

$$\alpha + e^{\varepsilon}\beta \geq 1-\delta, \quad \beta + e^{\varepsilon}\alpha \geq 1-\delta \tag{2}$$

hold for any adjacent dataset X, X' and any decision rule $\mathcal{R}$ in a hypothesis testing problem defined in Equation (1).

Theorem 1 bounds the testing region for any decision rule $\mathcal{R}$ if $\mathcal{M}$ is indeed differentially private. The basic principle of privacy audit is to output a privacy lower bound by contraposition of Theorem 1, i.e., if some achievable α, β falls out of the region defined by (ε, δ), it suggests that $\mathcal{M}$ is not (ε, δ)-DP. Roughly, the general procedure to produce a privacy lower bound is as follows.

After simulating the distinguishing game n times, for each simulation, the adversary makes a binary guess about which of two adjacent datasets was used. This gives the result of many pairs

$$\{(b_g, b_t)_i : i \in [n]\} \tag{3}$$

where b_g is the guessed result and b_t is the true secret for one simulation.

Under the empirical approach, we can derive confidence intervals for false positive rate $\alpha \in (\alpha_l, \alpha_r)$ and false negative rate $\beta \in (\beta_l, \beta_r)$ at some confidence level γ, based on $\{(b_g, b_t)_i : i \in [n]\}$ obtained. This is usually done by the Clopper-Pearson method [10] such that α and β are modeled as unknown success probabilities of two binomial distributions. And this leads to the privacy lower bound for the true privacy parameter ε_T at fixed δ according to Equation (2).

$$\varepsilon_T \geq \varepsilon_L = \max\{\log\frac{1-\delta-\alpha_r}{\beta_r}, \log\frac{1-\delta-\beta_r}{\alpha_r}, 0\} \tag{4}$$

Note that the privacy claim of the private algorithm reports an upper bound $\varepsilon_U \geq \varepsilon_T$ for some fixed δ.

2.3 Related Work

The whole picture of detecting privacy violations. On detecting privacy violations in the implementation of some differentially private algorithms, there are roughly two mainstream of work that rely on different techniques: 1) using formal verification methods to prove or disprove programs of DP algorithm [4, 17, 40, 49]; 2) generate refutation for targeted DP algorithms [7, 12, 20, 31, 37] based on statistical estimation.

The former often suffers from issues including being not applicable [40] to (ε, δ)-DP, Renyi-DP [29] or another advanced DP application called private selection [23]; some work also

requires necessary manual design assistance [49]. Another limitation is that those works cannot handle complex cases where the DP algorithm is part of some larger program [40].

In contrast, the latter technique, based on statistical estimation, is free from such issues. Therefore, it is more generalizable, although it may incur the heavy computational overhead of running the targeted DP algorithm thousands of times. Our work also falls under the latter category.

The statistic-estimation category. Privacy audit in this line of work can be framed as aiming to produce a privacy lower bound for the privacy parameter of the targeted algorithm. Certain earlier studies [6, 7, 12, 24] focus on generating privacy lower bound for some light-weight private protocols, including the Laplace mechanism or sparse vector techniques where running the targeted private algorithm is not a significant computational issue.

Privacy audit in machine learning tasks mainly focuses on investigating the theoretical versus practical privacy guarantees of the DP-SGD protocol [20, 21, 31]; one notable work is that Nasr et al. [31] show that the theoretical privacy analysis for DP-SGD is indeed tight.

Privacy audit in machine learning benefits from the following lines of related work:

- **Better membership inference**. Under the context of privacy audit, some form of membership inference [8, 34] needs to be instantiated in the distinguishing game. For example, Jagielski et al. [20] design worst-case data examples, a.k.a. "*canaries*" in literature, to form better membership inference, which leads to better privacy lower bound. This line of work aims to produce more informative $\{(b_g, b_t)_i : i \in [n]\}$ (Equation (3)) results of the distinguishing game.
- **Better estimation.** Work in this line aims to perform better statistical analysis over derived $\{(b_g, b_t)_i : i \in [n]\}$ (Equation (3)) results of the distinguishing game, and hence is independent of work on membership inference. For example, Log-Katz confidence interval [25] and advanced techniques based on Bayesian estimation [48] have been proposed to improve the final derived lower bound.

More efficient privacy audit. To address the possible computational issue of running the targeted private algorithm many times, improvements have been made on the "meta-level": arranging the membership inference and estimation to achieve auditing by fewer runs.

For instance, Nasr et al. [30] leverage the iterative structure of DP-SGD to perform the overall empirical privacy of DP-SGD; Andrew et al. [2] insert random canaries into the input to Gaussian mechanism and measure the privacy based on the result of recovering those random canaries *simultaneously*. Such heuristic of making multiple membership inferences per run of the targeted algorithm has also been used in previous works [27, 48] to improve the efficiency of privacy audit; however, such practices are without theoretical rigor; the final estimated privacy lower bound is also considered not faithful [48] because membership inferences are performed on data examples not belonging to independent runs, which invalidates current false positive/false negative rate estimation techniques.

Such a problem is further studied by a recent work by Steinke et al. [37] with theoretical justification. Steinke et al. [37] also propose an audit method that can derive the final privacy lower bound while requiring the targeted DP-SGD protocol to run only once.

2.4 Problem Statement and Motivation

To briefly describe the approach by Steinke et al. [37], 1) first, n contrived data examples (canaries) are decided to be included or not included in the training based on n independent coin flips; 2) second, perform membership inference on those n data examples based on the output of only one run of the targeted algorithm (DP-SGD); 3) finally, privacy lower bound is formed based on the accuracy of those membership inferences.

Problems and challenges. The central contribution made by Steinke et al. [37] is to validate the operation: performs membership inferences on multiple data examples not belonging to independent runs. However, significant problems with the privacy audit remain unanswered.

First, the audit method in [37] is not tight in general, e.g., it does not give tight privacy lower bound for the Gaussian mechanism, even when all parameters are carefully tweaked. Why does this happen, and can it be improved? Second, it is unclear how to transfer such a method to audit problems on DP algorithms other than the DP-SGD protocol. It is beneficial to have some principles to follow.

We also identify another question critical to the audit problem: since the targeted private algorithm is run only once and inserting more canaries becomes a cheap operation, if we ignore other considerations but only focus on deriving better privacy lower bounds, can we arbitrarily increase n to get arbitrarily high confidence for the lower bound estimation?

Remark. To our knowledge, Steinke et al. [37] are the first to provide privacy lower bound based on only one run of the targeted algorithm with theoretical rigor instead of heuristics. At the time of submission, their method is state-of-the-art. We also note another work [26] that improves over [37] based on [37]'s analysis, reaching slightly better audit results. However, [26]'s results are still not tight (in contrast, we reach tight results); more importantly, the above raised two problems still remain unanswered. In our experiment, we only compare with [37] as it suffices to show our contribution related to the above two problems.

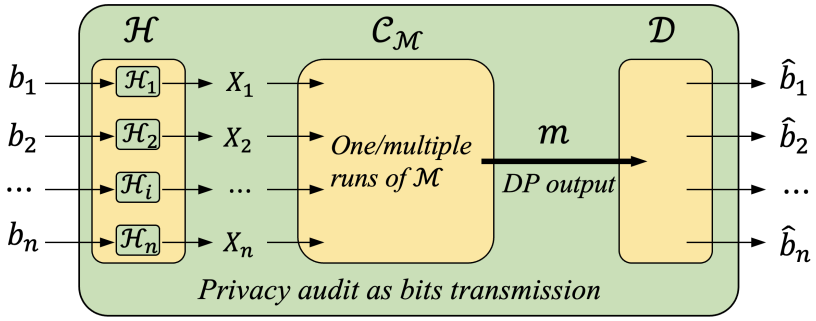


Figure 1: The universal framework for privacy audit. Each membership inference corresponds to recovering a bit. Execution of the targeted private algorithm $\mathcal{M}$ corresponds to the usage of a noisy channel for bits transmission. $\mathcal{C}_{\mathcal{M}}$ is the noisy channel where execution of $\mathcal{M}$ happens, and $\mathcal{D}$ is where the membership inference is launched. $\mathcal{H}$ is the dataset generator and m is what can be observed by the adversary.

3 Method

Method intuition. Addressing the above questions requires understanding the principle of privacy audit. In this section, we formulate the privacy audit problem as a *bits transmission* framework to serve such a goal. This builds on the observations that, in the membership inference, one of two adjacent datasets needs to be decided as the original input, and this is just equivalent to recovering one bit of information.

Based on our design framework, executing the targeted private algorithm $\mathcal{M}$ is modeled as a use of a noisy channel. If $\mathcal{M}$ is indeed DP, the channel will be noisy enough to prevent reliable bits transmission; therefore, if we can recover bits with low error, we can derive a privacy lower bound for $\mathcal{M}$.

Overview. We use the language of information theory to make such intuition precise. With such an analytical tool, 1) we derive an improved (tight) privacy audit method, which works for both cases of privacy-audit-by-one-run and privacy-audit-by-multiple-run; 2) we also analyze the fundamental limits of privacy audit, which tells us when the appealing audit-by-one goal is possible or impossible.

3.1 Universal Framework for Privacy Audit

Definition 5 (Privacy audit as bits transmission)**.** *The universal bit transmission framework $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ for privacy audit models a problem of n bits information transmitting through a noisy channel where*

1. *$b \in \{0,1\}^n$ is n-dimension binary vector where the i-th coordinate b_i of b is independently sampled from a Bernoulli distribution* **Bernoulli**(p) $\forall i \in \{1,2,\cdots,n\}$*;*

2. *Dataset generator $\mathcal{H} : \{0,1\}^n \to \mathcal{X}^n$ outputs an audit dataset $X_A = \{X_i = \mathcal{H}_i(b_i) : \forall i \in [n]\} \in \mathcal{X}^n$ where each data sample inside X_A only depends on the corresponding bit of $\mathcal{H}$'s input.*

3. *The noisy channel $\mathcal{C}_{\mathcal{M}} : \mathcal{X}^n \to I$ outputs message $m = \mathcal{C}_{\mathcal{M}}(X_A) \in I$. $\mathcal{C}_{\mathcal{M}}$ contains built-in information $\mathcal{M} : \mathcal{X}^* \to \mathcal{Y}$, the DP algorithm we want to audit. $\mathcal{C}_{\mathcal{M}}$ may also contain information of $\mathcal{M}$'s original (training) dataset that is independent of input bits b.*

4. *Decoder $\mathcal{D} : I \to \{0,1\}^n$ tries to recover information bits b but actually output $\hat{b} = \mathcal{D}(m) = \{b_i : \forall i \in [n]\} \in \{0,1\}^n$, which might result in errors.*

Regulation: *to comply with privacy audit, it is by design that 1) each data sample inside X_A must only be associated with exactly* ***one*** *run of $\mathcal{M}$; 2) message $m = \mathcal{C}_{\mathcal{M}}(X_A)$ must only be formed based on $\mathcal{M}$'s output, i.e., m is differentially private.*

Algorithm 1 $\mathcal{C}_{\mathcal{M}}$ for privacy audit by multiple runs

Input: X_A
1: $m \leftarrow []$
2: **for** $i = 1, \cdots, n$ **do**
3: Get original training dataset X_T
4: ▷ X_T can be different at each iteration
5: $y_i \leftarrow \mathcal{M}(\{X_A[i]\} \cup X_T)$ ▷ One run of $\mathcal{M}$
6: $m.append(y_i)$
7: **end for**
Output: m

Interpretation. To connect to previous privacy audit terminologies, in our $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework, the dataset generator $\mathcal{H}$ models the how the canary data examples are formed. The noisy channel $\mathcal{C}_{\mathcal{M}}$ models the execution of $\mathcal{M}$; being "noisy" corresponds to the fact that $\mathcal{M}$ hides the evidence of X_i's participation if $\mathcal{M}$ is indeed DP [39], which makes it hard to recover b_i. The decoder $\mathcal{D}$ is where the membership inference happens, such that a binary decision must be made for each input bit.

Handling privacy audit by multiple runs. For previous privacy audit work [20, 31, 48] falling under the category of privacy audit by multiple runs, these work can be framed by our $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework as follows.

The critical part is what happens inside the noisy channel $\mathcal{C}_{\mathcal{M}}$, for privacy audit by multiple runs, we demonstrate $\mathcal{C}_{\mathcal{M}}$ as shown in Algorithm 1. In this line of work, each membership inference is associated with one run of $\mathcal{M}$; here the transmission

$$b_i \to X_i \to y_i = m[i] \to \hat{b}_i$$

is equivalent to derive a $(b_i, \hat{b}_i) = (b_t, b_g)$ pair as defined in Equation (3). Note that all n runs of $\mathcal{M}$ are mutually independent.

Algorithm 2 $\mathcal{C}_{\mathcal{M}}$ for privacy audit by one run

Input: X_A

1: Get original training dataset X_T

2: $y \leftarrow \mathcal{M}(X_A \cup X_T)$ ▷ One run of $\mathcal{M}$

3: $m \leftarrow y$

Output: m

Handling privacy audit by one run. For previous audit work [37] falling under privacy audit by one run, such work can be framed by our $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework, shown in Algorithm 2. The critical part is that inside $\mathcal{C}_{\mathcal{M}}$, all generated data examples (corresponding to canaries) are all fed into the input dataset of $\mathcal{M}$ and $\mathcal{M}$ runs on them together only once. In this case, bit transmission may not necessarily be independent, i.e., there might be *interference* between them.

Handling two versions of adjacency. Our framework applies to audit tasks for both versions of the adjacency definition of DP. If the $\mathcal{M}$ is DP with respect to addition/removal, we require X_i to be either a real canary data example or a *null* object that contributes nothing to the execution of $\mathcal{M}$, based on b_i. This null setup models the case where the canary is not included in $\mathcal{M}$'s. If the $\mathcal{M}$ is DP with respect to replacement, X_i can be any different data examples depending on b_i.

Basic considerations. The final goal is to estimate the privacy lower bound of $\mathcal{M}$ based on $b = \{b_i : \forall i \in [n]\}$ and $\hat{b} = \{\hat{b}_i : \forall i \in [n]\}$. By basic design principles, as we can always let $\hat{b}$ be bad guesses (e.g., just make $\hat{b}$ random), it is pivotal for $\hat{b}$ to recover b as accurate as possible, which allows to conclude more informative assertion about $\mathcal{M}$'s privacy (deriving stronger privacy lower bound).

3.2 Information-theoretic Limits

In this section, we give results demonstrating the fundamental information-theoretic limits under our audit framework $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$. Those limits always hold if complying with our framework, regardless of whether the privacy audit is by one run or multiple run of $\mathcal{M}$.

Notation. We use uppercase letters (e.g., Z) to represent a random variable and lowercase letters (e.g., z) to denote its realization. When we need to refer to the distribution of Z, we also abuse using the uppercase without ambiguity, e.g.,

$$Z|_{Y=y} \tag{5}$$

denotes the distribution of random variable Z conditioned on random variable $Y = y$. Notations for random variables used are summarised in Table 1.

Notation	Meaning
B	Random input bits sampled from **Bernoulli**$(p)^n$
B_i	Marginal distribution of i-th coordinate of B
$\hat{B}$	Random recovered bits
$\hat{B}_i$	Marginal distribution of i-th coordinate of $\hat{B}$
M	Random variable for $\mathcal{C}_{\mathcal{M}}$'s output
E_i	Bit error random variable defined in Equation (6)
E	Bit error random vector $E = [E_1, \cdots, E_n]$
	For some random vector R
R_{-i}	Random vector $R_{-i} = [R_1, \cdots, R_{i-1}, R_{i+1}, \cdots R_n]$
$R_{<i}$	Random vector $R_{<i} = [R_1, \cdots, R_{i-1}]$
Lowercase use corresponds to the above's realization	

Table 1: Notation for random variables used.

We care about the errors made in bits transmission, which is formally defined in the following.

Definition 6. *Define a random variable E_i as follows.*

$$E_i = \begin{cases} 1, \text{ if } \hat{B}_i \neq B_i \\ 0, \text{ if } \hat{B}_i = B_i \end{cases} \tag{6}$$

I.e., E_i is the random variable indicating whether recovered bit $\hat{B}_i$ is an error. We denote

$$p_i^e = \Pr[E_i = 1] \tag{7}$$

*as the **bit error** for i. We also define the **average** bit error as*

$$p^e = \frac{1}{n}\sum_i \Pr[E_i = 1] = \frac{1}{n}\sum_i p_i^e \tag{8}$$

We will also formalize the indistinguishability provided by DP algorithm under our framework using f-DP. If distribution P, P' possessing some level of indistinguishability and their trade-off function satisfies

$$T_{P,P'} \geq f$$

we denote this relation between P, P' as

$$P \overset{f-\text{DP}}{\sim} P'. \tag{9}$$

It is easy to see that $\mathcal{M}$ is f-DP if $\mathcal{M}(X) \overset{f-\text{DP}}{\sim} \mathcal{M}(X')$ for all adjacent X, X'.

Implication from our framework regulation and differential privacy. With all the notations set, we are ready to state some facts based on our framework and differential privacy. We have the following property according to the regulation shown in Definition 5.

Property 1 (Noisy transmission implied by DP, proof in Appendix B.1)**.** *In our $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework, if $\mathcal{M}$ is f-DP, then $\forall i \in [n], b_{-i} \in \{0,1\}^{n-1}$*

$$\hat{B}_i|_{B_i=0, B_{-i}=b_{-i}} \overset{f\text{-}DP}{\sim} \hat{B}_i|_{B_i=1, B_{-i}=b_{-i}} \tag{10}$$

Where $[n] = \{1, 2, \cdots, n\}$. Equation (10) intuitively says that, conditioned on other input bits being fixed to be b_{-i}, even if we flip the i-th input bit, its corresponding output bit's distribution will not change much.

We present a lemma that we later rely on as follows.

Lemma 1 (Mixture by convex combination only makes it more indistinguishable, proof in Appendix B.2)**.** *If $P_i \overset{f\text{-}DP}{\sim} P_i', \forall i = 1, \cdots, n$, then $\forall c_i \in [0,1]$ such that $\sum_{i=1}^n c_i = 1$, we have*

$$\sum_{i=1}^n c_i P_i \overset{f\text{-}DP}{\sim} \sum_{i=1}^n c_i P_i'. \tag{11}$$

intuitively, the pair of corresponding mixture distributions by convex combination becomes only harder to distinguish than each P_i, P_i' pair.

Hardness of single bit transmission. Under our $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework, the marginal distribution for some single output bit also possesses some level of indistinguishability condition on different corresponding input bit.

Corollary 1 (Recovering bits is hard, proof in Appendix B.3)**.** *In our $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework, if $\mathcal{M}$ is f-DP, then $\forall i \in [n]$,*

$$\hat{B}_i|_{B_i=0} \overset{f\text{-}DP}{\sim} \hat{B}_i|_{B_i=1} \tag{12}$$

We need the help of Lemma 1 to complete the proof provided in Appendix B.3. Based on this result, we abstract the channel for the transmission of the i-th bit in Figure 2. We define the null hypothesis as $b_i = 0$ and the alternative hypothesis as $b_i = 1$. An arbitrary decision rule $\mathcal{R}$ makes a false negative rate β as a false positive rate α.

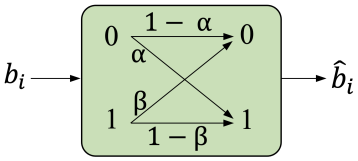


Figure 2: Single-bit transmission, modeled as a binary channel. If input bit $b_i = 0$, the channel flips the bit with probability α, corresponding to a false positive rate; if $b_i = 1$, the bit is flipped with probability β, which is the false negative rate. As governed by the trade-off function, $\beta \geq f(\alpha)$ must hold. if $\alpha = \beta$, the above channel is the well-known binary symmetric channel (BSC).

Hence, our channel molding tells us recovering the input bits is fundamentally hard due to the noisy channel $\mathcal{C}_{\mathcal{M}}$.

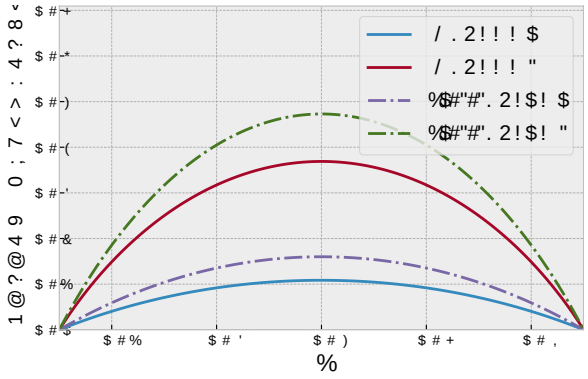


Figure 3: The mutual information upper bound $u_f(p)$ for different trade-off functions. $\delta = 10^{-5}$ for (ε, δ)-DP.

How do we quantify such hardness? We use the information-theoretic quantity: *mutual information*, a central topic in information theory. We use

$$\mathrm{MI}(B_i; \hat{B}_i) \tag{13}$$

to denote the mutual information between two binary random variables B_i and $\hat{B}_i$. A closely related quantify is the channel capacity at fixed α, β

$$\mathbb{C} = \max_p \mathrm{MI}(B_i; \hat{B}_i) \tag{14}$$

Under the original information-theoretic context, channel capacity is the maximum rate at which we can send information over some noisy channel with a vanishingly low error probability. Under our privacy audit context, it is related to the best bit error that can be achieved, as will be shown later.

Assuming we are using 2 as the base of logarithm for information related quantities throughout this paper, the binary entropy function is

$$\mathrm{h}(x) = -x \cdot \log(x) - (1-x) \cdot \log(1-x) \tag{15}$$

where $\mathrm{h}(x) : [0,1] \to [0,1]$ is symmetric around $x = \frac{1}{2}$. Also, define an inverse function

$$\mathrm{h}^{-1}(x) : [0,1] \to [0, \frac{1}{2}] \tag{16}$$

corresponding to the inverse of the left half part of $\mathrm{h}(x)$ defined on $x \in [0, \frac{1}{2}]$.

Intuitively, mutual information $\mathrm{MI}(B_i; \hat{B}_i)$ measures the dependence between random bits B_i and $\hat{B}_i$. In our $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework, note that $B_i \sim \textbf{Bernoulli}(p)$, if $\mathrm{MI}(B_i; \hat{B}_i)$ reaches to its maximum $\mathrm{MI}(B_i; \hat{B}_i) = \mathrm{H}(B_i) = \mathrm{h}(p)$ where $\mathrm{H}(B_i)$ is the entropy of B_i, we then have $\hat{B}_i = B_i$, i.e., perfect transmission.

However, the algorithm $\mathcal{M}$ prevents perfect transmission if it is private. We have the following results in quantifying the fundamental hardness of single-bit transmission.

Theorem 2 (Mutual information upper bound for bits transmission, proof in Appendix B.4)**.** *in our* $(n, p, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ *framework, if* $\mathcal{M}$ *is* f*-DP, we have* $\forall i \in [n]$

$$\mathrm{MI}(B_i; \hat{B}_i) \leq \max_{x \in [0,1]} F_f(x, p) \stackrel{def}{=} u_f(p) \tag{17}$$

where

$$\begin{aligned} F_f(x,p) =& \mathrm{h}(p \cdot f(x) + (1-p)(1-x)) - p \cdot \mathrm{h}(f(x)) \\ & - (1-p) \cdot \mathrm{h}(1-x) \end{aligned} \tag{18}$$

Proof of this theorem relies on our channel modeling shown in Figure 2. In practice, computing $u_f(p)$ is always numerically stable as all terms are bounded. Figure 3 plots the upper bound function $u_f(p)$, and we can see that the more private trade-off function tends to have a smaller upper bound value.

Equation (18) leads to a somewhat complex form because we allow p to be chosen freely. We can make it concise by setting $p = \frac{1}{2}$, i.e., each bit is independently and uniformly sampled from $\{0, 1\}$. This setup corresponds to the balanced prior adopted in previous work [31, 33, 34]. And we assume $p = \frac{1}{2}$ in the remaining part unless specified otherwise.

Theorem 2 also tells us that, even if the membership inference does its best, i.e., the false positive rate α and false negative rate $\beta = f(\alpha)$ lands on boundaries defined by the trade-off function, the mutual information still has an upper bound. Consequently, it leads to unavoidable non-trivial error in bits transmission, as stated by the following result.

Theorem 3 (Bit error lower bound, proof in Appendix B.5)**.** *In our* $(n, \frac{1}{2}, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ *framework, w.o.l.g., assume* $p_i^e \leq \frac{1}{2}$ *(defined in Equation (7)), because one can always do better than random guessing. If* $\mathcal{M}$ *is* f*-DP, we have* $\forall i \in [n]$

$$p_i^e \geq \mathrm{h}^{-1}(1 - u_f(\frac{1}{2})) \stackrel{def}{=} p_f^e \tag{19}$$

Where h^{-1} *is defined in Equation (16).*

Theorem 3 is the basis of our audit method shown in the following section.

3.3 Audit Method

Method overview. Our idea for privacy audit is fairly simple: if $\mathcal{M}$ is some f-DP, 1) for each bit transmission, the bit error will not be too small; 2) therefore, if we observe some significantly low bit error, we can conclude (with confidence specification) that $\mathcal{M}$ is not f-DP as claim by contraposition. This gives a privacy lower bound.

However, deriving a lower bound requires non-trivial manipulation. Our method in this section quantifies the relationship between bit error and the lower bound that we can conclude.

Privacy audit without interference. We first assume that each bit transmission $b_i \to \hat{b}_i$ is mutually independent. Under the information-theoretic context, this equals to the fact that there is *no interference* between bits transmission; another equivalent characterization is that the noisy channel $\mathcal{C}_{\mathcal{M}}$ is a *memoryless* channel.

For the privacy audit task design, whether interference exists between bits transmission can always be controlled. For example, we can always resort to the privacy-audit-by-multiple-run case where interference does not exist. Actually, for privacy audits, ensuring no interference should be viewed as a principle to follow. We will provide this argument in later sections, and we will also provide an analysis for the case where interference exists.

In the case where interference does not exist, the random variable E_i ((defined in Equation (6)) is independent of E_j, $\forall i, j \in [n]$ and $i \neq j$. Note that the error probability $\{p_i^e : \forall i \in [n]\}$ (defined in Equation (7)) may not necessarily be the same.

Theorem 4 (Audit principle, proof in Appendix B.6)**.** *In our* $(n, \frac{1}{2}, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ *framework, if* E_i *(defined in Equation (6)) is independent from* E_j, $\forall i, j \in [n]$ *and* $i \neq j$*, let* S *be a* n*-dimension binary vector where each coordinate of* S *is independently sampled from* **Bernoulli**(p_f^e) *(*p_f^e *defined in (19)). We have* $\forall a \in [0, 1]$

$$\Pr_E\left[\frac{1}{n}\sum_i^n E_i \geq a\right] \geq \Pr_S\left[\frac{1}{n}\sum_i^n S_i \geq a\right] \tag{20}$$

This theorem says that it is more likely to see a greater value for the average of results sampled from $\{E_i : \forall i \in [n]\}$ than that of results sampled from n independent Bernoulli tries with probability p_f^e. Intuitively, this is because the probability of making an error ($E_i = 1$) for each bit is greater than p_f^e, which is the best we can do. It allows us to make the following deductions, which also form the basis of our audit method.

Confidence interval (CI) construction. In our $(n, \frac{1}{2}, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework, if $\mathcal{M}$ is f-DP,

1. If we really can achieve the "best". The bit error random variable becomes $E_i = S_i \sim$ **Bernoulli**$(p_f^e), \forall i \in [n]$.;
2. For pre-defined confidence level γ (95% and 99% are typical), we can compute v such that
$$\Pr_S\left[\frac{1}{n}\sum_i S_i \geq p_f^e - v\right] = \gamma \tag{21}$$
3. For some real random variable E_i that we can actually achieve, Theorem 4 promises that
$$\Pr\left[\frac{1}{n}\sum_i E_i \geq p_f^e - v\right] \geq \gamma \tag{22}$$
4. Hence, once we observe an outcome $\bar{e}$ for the random variable of the sample mean $\frac{1}{n}\sum_i E_i$, Equation (22) give

Algorithm 3 Advanced CI $\mathcal{ACI}(\bar{e},\gamma,n)$

Input: $\bar{e}$, empirical average bit error; γ confidence specification; n, total number of bits transmission

$p_l \leftarrow 0.001$, $p_r \leftarrow \frac{1}{2}$
$p^e_{min} \leftarrow \frac{1}{2}(p_l + p_r)$
$\triangleright$ F^{-1} is the inverse c.d.f. of a Binomial distribution
$v \leftarrow p^e_{min} - \frac{1}{n}F^{-1}(1-\gamma, n, p^e_{min})$
for $\|p^e_{min} - (\bar{e}+v)\| > 0.0001$ **do**
  if $p^e_{min} > \bar{e} + v$ **then**
    $p^e_r \leftarrow p^e_{min}$
  else
    $p^e_l \leftarrow p^e_{min}$
  end if
  $p^e_{min} \leftarrow \frac{1}{2}(p_l + p_r)$
  $v \leftarrow p^e_{min} - \frac{1}{n}F^{-1}(1-\gamma, n, p^e_{min})$
end for

Output: p^e_{min}

us the confidence interval

$$[0, \bar{e} + v] \tag{23}$$

for p^e_f (the parameter we aim to estimate) with a coverage level always greater than γ.

The remaining task is how to compute v and we can let $v = v(n,\gamma)$, a function of n,γ, which can be tackled by the Hoeffding bound as follows

$$v = \sqrt{\frac{1}{2n}\log\frac{1}{1-\gamma}} \tag{24}$$

The derivation for Equation (24) is a standard application of Hoeffding's inequality, and it is presented in Appendix A.2.

A more sophisticated CI construction method. Computing v by Equation (24) is simple, however, we can do better. For completeness, we also provide another more sophisticated CI construction method in the following, but paying the price for a slightly higher complexity.

The general idea is to let $v = v(p^e_f, n, \gamma)$, i.e., it also depends on p^e_f, and we can derive better results for v by iteration. The high-level intuition and procedure are as follows. After seeing $\bar{e}$ (which we cannot control in estimation), we want v as small as possible because it will lead to better privacy lower bound. Then, we can compute v by Equation (21) based on Binomial distribution; we repeat such process by setting different hypothetical values for $p^e_f \leftarrow p^e_{min}$ until a certain condition is met. The detailed method is provided in Algorithm 3.

Deriving the final privacy lower bound. we can derive the final privacy lower bound after the confidence interval is constructed. The lower bound is in (ε,δ)-DP formulation, aligning with almost all previous work. The whole process is presented in Algorithm 5. Note that in our method, we

Algorithm 4 f-DP to (ε,δ)-DP [13] $\mathcal{ED}(f,\delta)$

Input: f, trade-off function; δ, privacy parameter

$\varepsilon \leftarrow \infty$ if $\delta < 1 - f(0)$; Return ∞
Compute $\varepsilon = \inf\{a : f(x) \geq 1 - \delta - e^a x, \forall x \in [0,1]\}$ via binary search

Output: $\max\{0, \varepsilon\}$

Algorithm 5 Privacy lower bound $\mathcal{LB}(\delta,\bar{e},\gamma,n)$

Input: δ, privacy parameter; $\bar{e}$, empirical average bit error; γ confidence specification; n, total number of bits transmission

$p^e_f \leftarrow \sqrt{\frac{1}{2n}\log\frac{1}{1-\gamma}}$ or $p^e_f \leftarrow \mathcal{ACI}(\bar{e},\gamma,n)$ $\triangleright$ Algorithm 3
Compute f s.t. $\mathrm{h}(p^e_f) = 1 - u_f(\frac{1}{2})$ $\triangleright$ Equation (19)
$\varepsilon_L \leftarrow \mathcal{ED}(f,\delta)$ $\triangleright$ Algorithm 4

Output: ε_L

essentially estimate an upper bound for averaged bit error, which corresponds to an upper bound of f-DP, which can be converted into a final privacy lower bound ε_L. The conversion is presented in Algorithm 4.

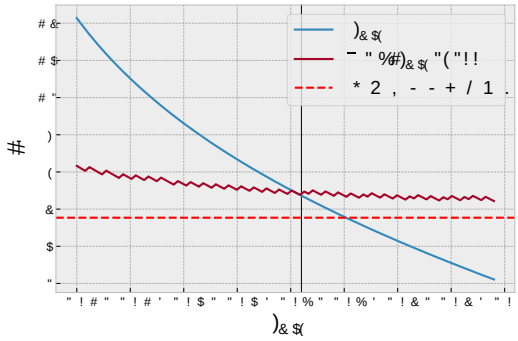
Understanding the advanced CI method. After introducing how the final lower bound is derived, we elaborate on how our advanced technique works. Recall that we set different hypothetical values p^e_{min} to p^e_f; each time we set a value to p^e_f, we are essentially making an assumption on the privacy lower bound. Specifically, if we assume we can achieve average bit error p^e_{min}, we are assuming the privacy lower bound derived based on this assumption is at least some value $\varepsilon_L(p^e_{min})$ where $\varepsilon_L(p^e_{min})$ is computed by the last two lines in Algorithm 5 by setting $p^e_f \leftarrow p^e_{min}$.

Based on assuming that p^e_{min} is the best we can achieve, we can derive the upper bound $\bar{e} + v(p^e_{min}, n, \gamma)$ of the confidence interval. If $\varepsilon_L(\bar{e} + v(p^e_{min}, n, \gamma)) < \varepsilon_L(p^e_{min})$, we have a contradiction, which requires we revise the assumption p^e_{min} until there are no more contradictions. The illustration is provided in Figure 4. Our advanced method has a notable advantage over the Hoeffding method when we do not have a large number of observations (n is small).

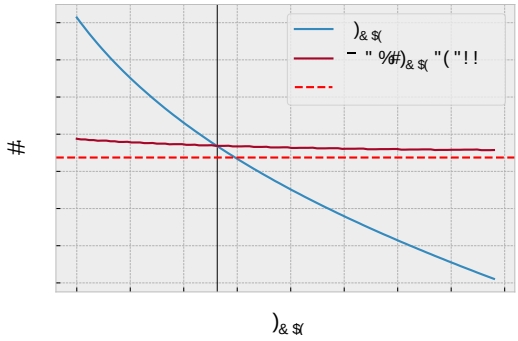
3.4 Principle for Channel Arrangement

In this section, we will justify why arranging the channel $\mathcal{C}_{\mathcal{M}}$ to be a memoryless channel for bits transmission is always superior to bits transmission with interference. In our $(n,p,\mathcal{H},\mathcal{C}_{\mathcal{M}},\mathcal{D})$ framework, we will also see the fact that avoiding interference should stand as an unquestioned design principle for all audit tasks.

Performance metric and sub-channel arrangement. To illustrate both cases for memoryless channel arrangement and channel with interference, we show an example in Figure 5 when $n = 2$. When there is no interference, we use $\mathbf{P}_{\hat{B}_i|B_i}$ to

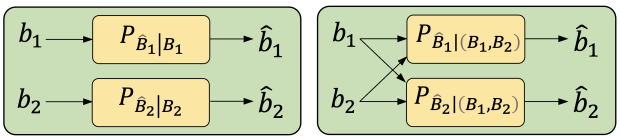


(a) $\bar{e} = 0.2, n = 100, \gamma = 0.99$



(b) $\bar{e} = 0.2, n = 10^3, \gamma = 0.99$

Figure 4: Illustration on how our advanced CI method works. The horizontal axis is different p^e_{min} value we assume that we can achieve, and the vertical axis is the corresponding lower bound. Line marked as p^e_{min} corresponds to lower bound derived based on average bit error p^e_{min} can be achieved; the same is to $\bar{e} + v(p^e_{min}, n, \gamma)$. Hoeffding result is the simple CI result shown in Equation (24). Regions on the left of the vertical black line are where we have contradictions.



(a) Memoryless channel (b) Channel with interference

Figure 5: Diagram of the case where we have $n = 2$ bits of transmission. When there is interference, output bit $\hat{b}_1$ also depends on input b_2, but $\hat{b}_1$ is only intended to recover b_1.

denote the transition probability matrix or Markov kernel for the channel where B_i and $\hat{B}_i$ are the input and output random variable, respectively. Such Markov kernel models the overall effect due to $\mathcal{H}$, $\mathcal{C}_{\mathcal{M}}$ and $\mathcal{D}$. In the presence of interference, the output random variable $\hat{B}_i$ depends on both inputs. Hence we have the Markov kernel denoted by $\mathbf{P}_{\hat{B}_l|(B_i,B_2)}$.

Practitioners can freely choose these two cases within our framework; however, as will be shown, the memoryless channel arrangement is more favored for our auditing purposes. To compare them, we need a performance metric. It is related to the following critical terms in our audit method.

Recall our previously mentioned audit method requires deriving an observation $\bar{e}$ for the random variable $\frac{1}{n}\sum_i E_i$, which is the sample mean. It inevitably drives us to assess how "good" we can make such an observation and what it implies to our final derived privacy lower bound. This is done via the following bias and variance argument.

❶ **The bias argument.** In Equation (8), we define the *averaged bit error* p^e which is the expectation for $\frac{1}{n}\sum_i E_i$. The bias term describes the distance between p^e achieved in an audit implementation and the best achievable bit error. We can trivially derive the result $p^e = \sum_i p^e_i \geq p^e_f$ as a result of Theorem 3. We want p^e as small as possible by the basic considerations mentioned in Section 3.1. Now a question arises: when and how p^e equals p^e_f.

The following theorem result gives us a more informative result for such a question.

Theorem 5 (Achievability of p^e_f, proof in Appendix B.7)**.** *In our $(n, \frac{1}{2}, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework, with p^e_i defined in Equation (7), if $\mathcal{M}$ is f-DP, then we have $\forall i \in [n]$,*

$$1 - \mathrm{h}(p^e_i) \leq \mathrm{MI}(B_i; \hat{B}_i) \overset{\mathbf{1}}{\leq} \mathrm{MI}(B_i; \hat{B}_i | B_{-i}) \leq u_f(\frac{1}{2}) \tag{25}$$

inequality **1** *becomes equality only when* $\mathrm{MI}(B_{-i}; \hat{B}_i) = 0$.

$\mathrm{MI}(B_{-i}; \hat{B}_i) = 0$ means recovered bit $\hat{b}_i$ is independent of input bits other than the intended b_i itself (i.e., there is no interference). Note that we can also derive Equation (19) based on the above result; nevertheless, the important fact told by Theorem 5 is that if $\mathcal{M}$ is f-DP, no matter how powerful the decoder $\mathcal{D}$ (where the MIA happens) is, p^e_f is not achievable for p^e in the presence of interference. In other words, a **non-zero bias** always exists between p^e and p^e_f.

This violates the basic design considerations mentioned in Section 3.1. Based on the bias argument, arranging bits transmission in a memoryless channel is better than a channel with interference.

❷ **The variance argument.** Our variance argument investigates the variance of the random variable $\frac{1}{n}\sum_i E_i$, which directly related to our confidence in our estimation. We want as low uncertainty (low variance in estimation) as possible, which applies to any other statistical estimation method.

In the case of a memoryless channel, E_i is independent of E_j for all $i \neq j$, which means that

$$\mathbf{Var}\left[\frac{1}{n}\sum_i E_i\right] = \frac{\sum_i \mathbf{Var}[E_i]}{n^2} \geq \frac{p^e_f(1-p^e_f)}{n} = V_{min} \tag{26}$$

Because $p^e_i = \Pr[E_i = 1] \geq p^e_f$ implies $\mathbf{Var}[E_i] \geq p^e_f(1-p^e_f)$.

Therefore, we favor the memoryless channel arrangement as it is only possible to achieve the best variance V_{min}.

Conclusion. Relating our audit mentioned before, we need the critical term $\bar{e}$ Equation (23) to compute a final lower bound. To have a non-trivial lower bound, we need 1) $\bar{e}$ to be as close

to p_f^e as possible (in expectation), i.e., we want low bias; 2) we need to have lower uncertainty in the estimation, i.e., $\bar{e}$ having small variance so that we can have non-trivial confidence in our estimation. A memoryless channel arrangement is more favored based on both lenses.

3.5 How the Decoder Affects Audit

The decoder $\mathcal{D}$ is where the membership inference attack happens, and we discuss how it affects the audit in the following. We can quantitatively reason about decoder $\mathcal{D}$, for instance, considering the Markov chain $B_i \to m \to \hat{B}_i$, $\mathcal{D}$ happens under transition $m \to \hat{B}_i$. Note that $\mathrm{MI}(B_i;\hat{B}_i) = \mathrm{MI}(B_i;m) - \mathrm{MI}(B_i;m|\hat{B}_i)$. Powerful $\mathcal{D}$ leads to $\mathrm{MI}(B_i;\hat{B}_i) = \mathrm{MI}(B_i;m)$ ($\mathrm{MI}(B_i;m|\hat{B}_i) = 0$), meaning that $\mathcal{D}$ may be a one-to-one mapping or sufficient statistics, allowing tight audit. Weak $\mathcal{D}$ leads to $\mathrm{MI}(B_i;\hat{B}_i) < \mathrm{MI}(B_i;m) \leq u_f(1/2)$ ($\mathrm{MI}(B_i;m|\hat{B}_i) > 0$), leading to non-zero bias, which must end up with a gap between the lower and upper bound for any auditing method.

Various factors may cause $\mathcal{D}$ to be weak: the membership inference attack is just sub-optimal, or there exist random sources unknown to the adversary (just like the adversary doesn't know which other data examples are sampled in each iteration of DP-SGD). Intuitively, $\mathcal{D}$ being weak means we lose information when processing the data. Once we know $\mathrm{MI}(B_i;m|\hat{B}_i)$, we know how quantitatively $\mathcal{D}$ affects the audit's tightness by Theorem 5, however, computing the value for $\mathrm{MI}(B_i;m|\hat{B}_i)$ should depend on the applications.

We also emphasize that in the presence of interference discussion in the above section, $\mathrm{MI}(B_i;\hat{B}_i) < \mathrm{MI}(B_i;m) \leq u_f(1/2)$ will also be true, causing the decoder $\mathcal{D}$ to be weak. This gives another reason why we should have a memoryless channel arrangement.

4 Privacy Audit by One Run: (Im)possibilities

Now, we are prepared to answer previously raised questions about the main topic we aim to discuss: *privacy audit by one run*.

The nature of privacy audit is to estimate the randomness due to DP. Using our bias and variance argument mentioned before, in all statistical estimation tasks, we always favor the expectation that the subject measured is close (lower bias) to its true value and high confidence (low variance). Under the context of privacy audit, what we really want to estimate is the *randomness injected by DP, which is parameterized by privacy parameters*.

For example, for the Gaussian mechanism

$$\mathcal{M}(X) = q(X) + \mathcal{N}(0, \sigma^2 \mathbb{I}^d) \tag{27}$$

where the query function $q(X) \in \mathbb{R}^d$ has unit ℓ_2-sensitivity, it is known that it satisfies (ε, δ)-DP if $\sigma^2 \geq 2\log(1.25/\delta)/\varepsilon^2$. Estimating a lower bound ε_L for ε is equivalent to estimating an upper bound for the noise s.t.d. σ. Inevitably, we must have enough observations of *independent* samples from the DP randomness itself before confidently claiming something about σ.

In privacy audit, the observations we have is the $\{(b_t, b_g)_i : \forall i \in [n]\}$ pairs of truth and guesses (Equation (3)). The goal is to estimate the randomness of DP based on those pairs. Interference will always lead to sub-optimal results based on our bias and variance argument.

When it is possible to audit privacy by one run. Suppose we plan to insert n canaries for audit. Privacy audit by one run is only possible if 1) the targeted private algorithm $\mathcal{M}$ itself incurs sampling from at least n independent DP randomness source; 2) n is large enough to have concentration behaviors.

The second requirement is because we need to have non-trivial confidence due to statistical uncertainties, and the first requirement is to have quality estimation based on our bias and variance argument. In the following, we will use a positive and negative example to give more insight into such necessary conditions for privacy audits by one run.

Example. In our audit framework, we aim to audit the privacy of the Gaussian mechanism defined in Equation (27). And the query function q is just a summation query

Positive case. If $d \geq n$, we can audit privacy by one run of the Gaussian mechanism and maintain the more favored memoryless channel arrangement mentioned previously. The canary data example is formed as each canary data $X_i \in R^d$ takes the value of 1 only its i-th coordinate and zero for the rest. The decoder $\mathcal{D}$ is also simple: for bit b_i, the recovered $\hat{b}_i$ is only based on the i-th coordinate of the output of $\mathcal{M}$. We can see that recovering $\hat{b}_i$ is free from other input bits.

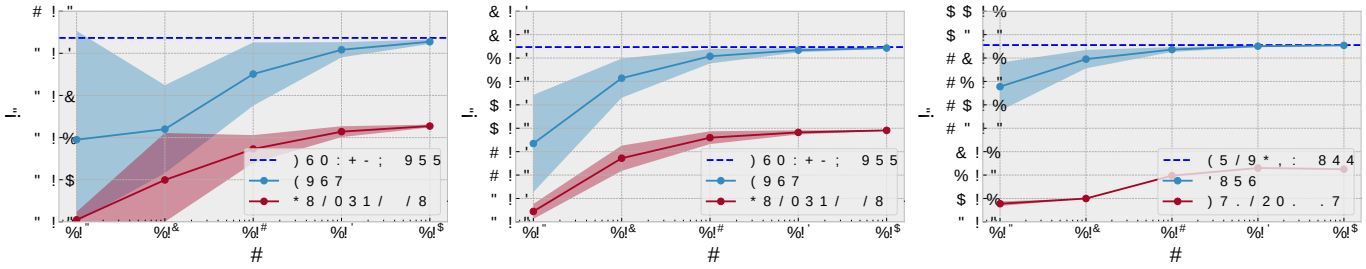
Negative case. If $d \ll n$, inserting more than d canaries will only lead to sub-optimal results according to our previous bias and variance argument, as we only have d independent source for DP noise. If d is too small, we can not have non-trivial confidence in our estimation. Therefore, audit-by-one-run is impossible for this case, so we have to resort to audit-by-multiple-runs to give meaningful privacy lower bound.

5 Experiments for Privacy Audit by One Run

5.1 Tight Audit by One Run

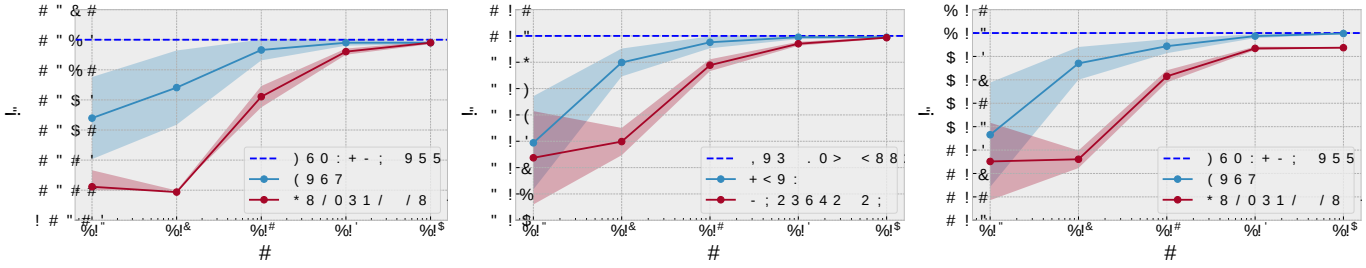
In this section, we first give results showing that our audit method is indeed tight. The confidence γ is set to be 0.95 throughout our experiments. We provide three experiments in the following.

How the experiments fits into our $(n, \frac{1}{2}, \mathcal{H}, \mathcal{C}_{\mathcal{M}}, \mathcal{D})$ framework. We follow identical setups as that of [37]. For Gaussian mechanism, in Equation (27), X contains rows with different one-hot vectors (row number one is [1,0,0,...,0], row number two is [0,1,0,...,0], etc); q sums up all vectors into one vector; $\mathcal{D}$ looks into each coordinate of final noisy vector (after Gaussian noise added) and output each $\hat{b}_i$. For auditing the Laplace



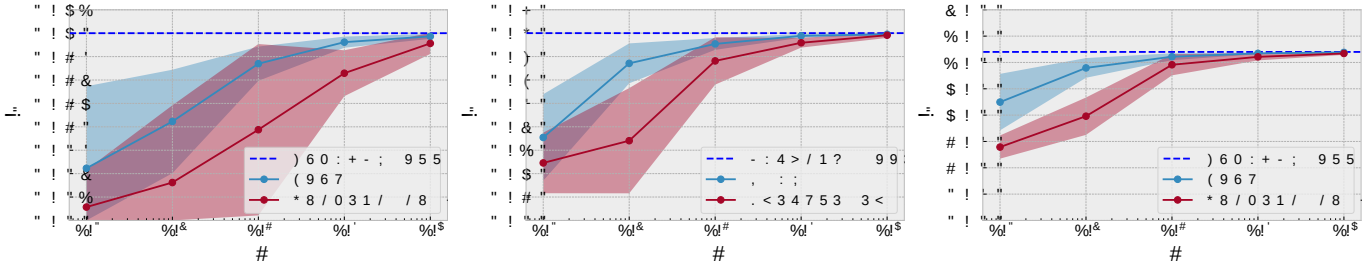
(a) $\mu = 0.2$ (b) $\mu = 0.8$ (c) $\mu = 3.2$

Figure 6: Audit by one run for the Gaussian mechanism satisfying μ-GDP. 20 repetition with different seeds.



(a) $\varepsilon = 0.25$ (b) $\varepsilon = 1$ (c) $\varepsilon = 4$

Figure 7: Audit by one run for the randomized response mechanism satisfying $(\varepsilon, 10^{-5})$-DP. 20 repetition with different seeds.



(a) $\mu_l = 0.2$ (b) $\mu_l = 0.8$ (c) $\mu_l = 3.2$

Figure 8: Audit by one run for the Laplace mechanism satisfying $T_{Lap(0,1),Lap(\mu_l,1)}$-DP. 20 repetition with different seeds.

mechanism in the following, the noise is merely changed to Laplace noise compared to auditing Gaussian mechanism. For randomized response, the DP randomness is randomly flipping the bits (instead of adding Gaussian noise) and $\mathcal{D}$ output each $\hat{b}_i$ based on noisy bits.

For example, in auditing the Gaussian mechanism, X_i refers to the i-th row of X, m is the final vector with Gaussian noise added, and $\hat{b}_i$ is the guessed bit for b_i. The dataset generator $\mathcal{H}$ is also simple: depending on b_i, it sets X_i to be a one-hot vector or a zero vector.

For asserting $\hat{b}_i$, we simply make decoder output 1 if the observed value is greater than 0.5 (the "threshold") for the Gaussian and Laplace mechanism. In previous work [30, 37], it is often to tune the threshold to reach the strongest audit performance. However, it suffices to set the threshold to be 0.5 in our case. For auditing the randomized response mechanism, we simply make the decoder output bit equal to the observed bit as it maximizes the posterior probability.

1) audit the Gaussian mechanism (Equation (27)) where we show our method obtains tight results and previous work [37] does not in contrast.

Setup. We aim to transmit n bits, and we let $d = n$. We vary n to take multiple values; the data example canary is according to the *positive case* in Section 4. This means that we have a memoryless channel arrangement. In this experiment, the query function q is just a summation query. In f-DP formulation, the Gaussian mechanism satisfies μ-GDP. We also vary μ to see audit results in different setups.

Results. Figure 6 shows the audit results using our method and the previous method by Steinke et al. [37]. And we can see that our method can achieve almost tight audit results. In contrast, the previous method cannot achieve tight results, as reported in the original work [37]. We believe one important reason is that [37] is based on (ε, δ)-DP formulation, which

is not tight/faithful [13, 51] for Gaussian mechanism.

By using f-DP formulation, we get tight results. It should also be noted that it is unclear how to transfer [37]'s result to handle the f-DP formulation.

2) audit randomized response mechanism [41] in an idealized setting where [37] gives tight audit results, but we show that our gives tight results with n less than one order of magnitude.

Setup. In our experiment, we have n bits to transmit, and randomized response turns the original bit into three possible outcomes: if $b_i = 0$, with probability $\frac{(1-\delta)e^\varepsilon}{1+e^\varepsilon}$, output 0; with probability $\frac{(1-\delta)}{1+e^\varepsilon}$, output 1; with probability δ, output 2. If $b_i = 1$, with probability $\frac{(1-\delta)e^\varepsilon}{1+e^\varepsilon}$, output 1; with probability $\frac{(1-\delta)}{1+e^\varepsilon}$, output 0; with probability δ, output 3. Then $\hat{b}_i$ is guessed based on such output. It is clear that such a mechanism satisfies (ε, δ)-DP. It is also clear that we have a memoryless channel arrangement.

Results. Figure 7 shows the audit results using our method and the previous method by Steinke et al. [37]. We see that the previous method can achieve tight audit results when $\varepsilon = 0.25, 1$, but a notable gap is still seen when $\varepsilon = 4$. In contrast, our method achieves tight results for all setups, and we obtain tight results with n being less by one order of magnitude.

3) audit the Laplace mechanism. The Laplace mechanism is summarized below:

$$\mathcal{M}(X) = q(X) + \mathcal{LAP}(0, c).$$

Note that $q(X) \in \mathbb{R}^d$ has bounded l_1-norm and $\mathcal{LAP}(0,c)$ is a d-dimension Laplace noise vector (with mean equal to zero and scale parameter c) where each coordinate is independent of each other.

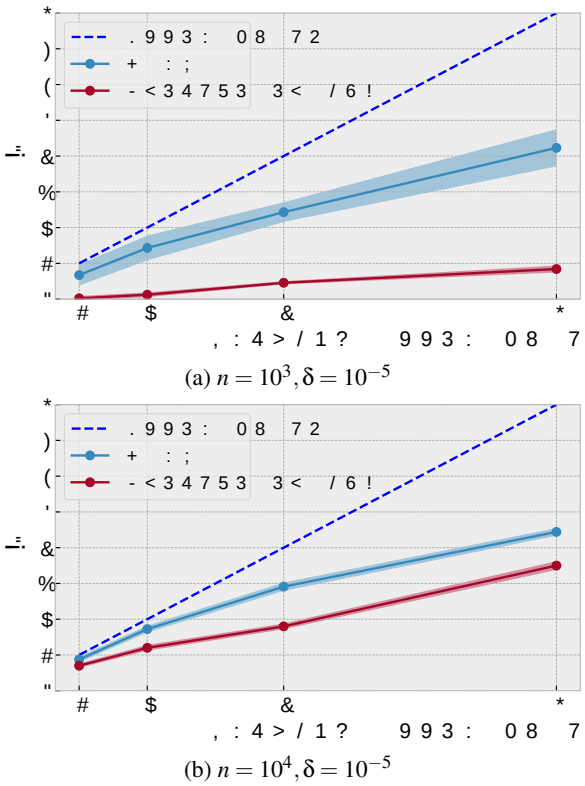
Setup. We set $q(X) \in \mathbb{R}^d$ has bounded l_1-norm equals to 1, then the mechanism satisfies $(\varepsilon_L = 1/c, 0) - DP$. By re-parameterizing, the trade-off function is $T_{Lap(0,1),Lap(\mu_l,1)}$-DP where $\mu_l = 1/c$.

Results. The results are presented in Figure 8. We can see that both audit methods can achieve tight results when $n = d$ is large enough; however, to reach the same lower bound, our method is more efficient by around one order of magnitude. This conclusion is similar to what Figure 7 tells us.

5.2 Experiments on DP-SGD

Our main contribution in this paper is analyzing the result of membership inference, particularly for the privacy-audit-by-one-run case; therefore, we do not focus on how to launch stronger membership inference, and we leverage previous techniques for membership inference. Both our method and previous method [37] are based on the same membership inference result, allowing us to have fair comparisons. Experiment implementation is at a link [1].

[1] https://github.com/zihangxiang/PAABT.git



(a) $n = 10^3, \delta = 10^{-5}$

(b) $n = 10^4, \delta = 10^{-5}$

Figure 9: Privacy lower bounds for auditing DP-SGD at white-box setting. The experimental dataset is CIFAR10, the same as that in [37].

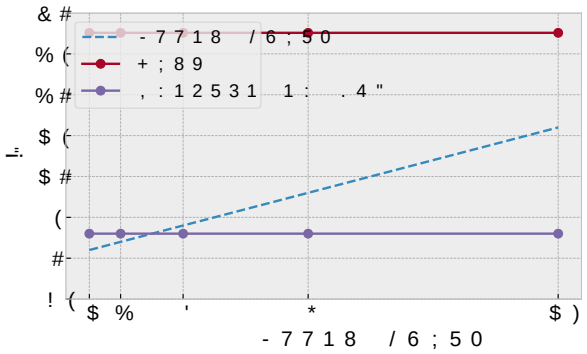
In this section, the audit results for DP-SGD protocols are given. We focus on the white-box setting where the intermediate private gradient is released publicly. We also leverage the membership inference method provided by Nasr et al. called "Dirac gradient," which directly inserts gradient candies where only one coordinate is 1 with others being zero. Such practice is similar to our above audit on the Gaussian mechanism, and we indeed have a memoryless channel arrangement based on a similar argument.

Results. Figure 9 presents the audit result. We can see that our method produces better lower bounds in each setting. Although the audit is not tight when performed on real-world training tasks when $n = 10^4$, both our method and the previous method give meaningful lower bounds; however, our method has significant advantages when $n = 10^3$.

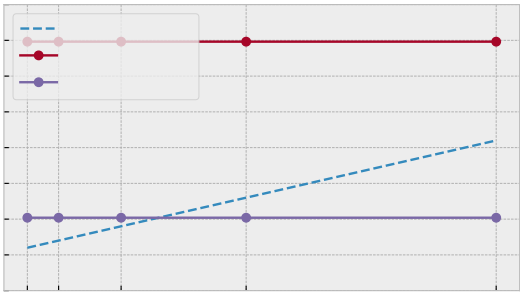
5.3 Detecting Privacy Violation

In this experiment, we provide a use case showing that our method catches bugs in real-world applications of differential privacy. This study is based on a pitfall in trying to refine the DP-SGD protocol. We briefly describe the root cause of such error made in [38] in the following.

The original DP-SGD protocol can be concisely summa-



(a) $n = 10^3, \delta = 10^{-5}$



(b) $n = 10^4, \delta = 10^{-5}$

Figure 10: Auditing a flawed private algorithm implementation.

rized as

$$p_i = \frac{1}{|\mathbf{B}|} \sum_i \text{CLP}_C(\nabla \ell_i) + \textbf{Noise}$$

where $\nabla \ell_i$ is the per-example gradient,$|\mathbf{B}|$ is the batchsize and $\text{CLP}_C(u) = u \cdot \min(1, \frac{C}{\|u\|_2})$ is a clipping operation. The final gradient p_i is differential private by adding calibrated noise. However, [38]'s solution accidentally turns the above protocol into

$$p_i = \sum_i \text{CLP}_C\left(\frac{1}{|\mathbf{B}|} \nabla \ell_i\right) + \textbf{Noise}$$

which makes the added noise to be considerable underestimated. We also focus on the white-box setting for auditing such privacy bugs.

Results. Figure 10 presents the audit results by both methods. Our method produces a rather strong lower bound, asserting that the algorithm is blatantly not as private as claimed under all privacy setups. In contrast, the previous method [37] only captures privacy violations for setups where the claimed privacy upper bounds are some small value and fails to catch the violations when they become greater.

6 Conclusion

In this work, we present a unifying framework that models the privacy audit problem through the lens of information-theoretic bit transmission. Using this approach, we derive fundamental limits for privacy audits and develop an audit method that provides tight empirical privacy bounds. Our framework also facilitates a detailed exploration of auditing via a single run of the target algorithm, investigating its feasibility and inherent limitations. These contributions not only enhance the accuracy and efficiency of privacy audits but also offer deeper insights into the problem of when and how auditing by one run is possible/impossible.

Our experimental results demonstrate that the proposed methods yield tighter privacy bounds with fewer (up to one order of magnitude) observations while also identifying privacy vulnerabilities in flawed differentially private implementations. These findings have important implications for strengthening privacy guarantees in real-world applications.

7 Ethics Considerations

We investigate the privacy guarantees of differentially private algorithms by developing and applying audit techniques designed to verify the privacy claims of these algorithms. Our method does not instantiate any real-world privacy attacks. The techniques and findings presented in this paper are intended solely for academic purposes and to advance the field of differential privacy. We aim to contribute to developing more secure and trustworthy privacy-preserving technologies.

8 Open Science

Our paper complies with the CFP open science policy. We follow the policy to openly share the research artifacts, including the source codes to realize secure quantile summary aggregation, scripts to run the evaluations, and documents to understand the functions of code with respect to the descriptions in this paper.

9 Acknowledgments

Di Wang and Zihang Xiang are supported in part by the funding BAS/1/1689-01-01, URF/1/4663-01-01, REI/1/5232-01-01, REI/1/5332-01-01, and URF/1/5508-01-01 from KAUST, and funding from KAUST - Center of Excellence for Generative AI, under award number 5940. Tiaohao Wang is supported by NSF CNS-2220433.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Galen Andrew, Peter Kairouz, Sewoong Oh, Alina Oprea, H Brendan McMahan, and Vinith Suriyakumar. One-shot empirical privacy estimation for federated learning. *arXiv preprint arXiv:2302.03098*, 2023.

[3] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in neural information processing systems*, 31, 2018.

[4] Gilles Barthe, Rohit Chadha, Vishal Jagannath, A Prasad Sistla, and Mahesh Viswanathan. Deciding differential privacy for programs with finite inputs and outputs. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 141–154, 2020.

[5] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

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

[7] Benjamin Bichsel, Samuel Steffen, Ilija Bogunovic, and Martin Vechev. Dp-sniper: Black-box discovery of differential privacy violations using classifiers. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 391–409. IEEE, 2021.

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

[9] Yan Chen and Ashwin Machanavajjhala. On the privacy properties of variants on the sparse vector technique. *arXiv preprint arXiv:1508.07306*, 2015.

[10] Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

[11] Meng Ding, Mingxi Lei, Liyang Zhu, Shaowei Wang, Di Wang, and Jinhui Xu. Revisiting differentially private relu regression. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[12] Zeyu Ding, Yuxin Wang, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. Detecting violations of differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 475–489, 2018.

[13] Jinshuo Dong, Aaron Roth, and Weijie Su. Gaussian differential privacy. *Journal of the Royal Statistical Society*, 2021.

[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[15] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 381–390, 2009.

[16] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[17] Gian Pietro Farina. *Coupled relational symbolic execution*. PhD thesis, State University of New York at Buffalo, 2020.

[18] Lijie Hu, Ivan Habernal, Lei Shen, and Di Wang. Differentially private natural language models: Recent advances and future directions. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 478–499, 2024.

[19] Lijie Hu, Zihang Xiang, Jiabin Liu, and Di Wang. Privacy-preserving sparse generalized eigenvalue problem. In *International Conference on Artificial Intelligence and Statistics*, pages 5052–5062. PMLR, 2023.

[20] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33:22205–22216, 2020.

[21] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[22] PRNG key reuse in differential privacy. https://github.com/google/jax/pull/3646.

[23] Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 298–309, 2019.

[24] Johan Lokna, Anouk Paradis, Dimitar I Dimitrov, and Martin Vechev. Group and attack: Auditing differential privacy. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1905–1918, 2023.

[25] Fred Lu, Joseph Munoz, Maya Fuchs, Tyler LeBlond, Elliott Zaresky-Williams, Edward Raff, Francis Ferraro, and Brian Testa. A general framework for auditing differentially private machine learning. *Advances in Neural Information Processing Systems*, 35:4165–4176, 2022.

[26] Saeed Mahloujifar, Luca Melis, and Kamalika Chaudhuri. Auditing f-differential privacy in one run. *arXiv preprint arXiv:2410.22235*, 2024.

[27] Mani Malek Esmaeili, Ilya Mironov, Karthik Prasad, Igor Shilov, and Florian Tramer. Antipodes of label differential privacy: Pate and alibi. *Advances in Neural Information Processing Systems*, 34:6934–6945, 2021.

[28] Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 650–661, 2012.

[29] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[30] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. In Joseph A. Calandrino and Carmela Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pages 1631–1648. USENIX Association, 2023.

[31] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pages 866–882. IEEE, 2021.

[32] Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

[33] Ahmed Salem, Giovanni Cherubin, David Evans, Boris Köpf, Andrew Paverd, Anshuman Suri, Shruti Tople, and Santiago Zanella-Béguelin. Sok: Let the privacy games begin! a unified treatment of data inference privacy in machine learning. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 327–345. IEEE, 2023.

[34] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[35] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pages 245–248. IEEE, 2013.

[36] Thomas Steinke. Composition of differential privacy & privacy amplification by subsampling. *arXiv preprint arXiv:2210.00597*, 2022.

[37] Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy auditing with one (1) training run. *arXiv preprint arXiv:2305.08846*, 2023.

[38] Timothy Stevens, Ivoline C Ngong, David Darais, Calvin Hirsch, David Slater, and Joseph P Near. Backpropagation clipping for deep learning with differential privacy. *arXiv preprint arXiv:2202.05089*, 2022.

[39] Michael Carl Tschantz, Shayak Sen, and Anupam Datta. Sok: Differential privacy as a causal property. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pages 354–371. IEEE, 2020.

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

[41] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[42] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

[43] Wikipedia. Hoeffding's inequality — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Hoeffding's%20inequality&oldid=1241370157, 2024.

[44] WSJ. https://www.wsj.com/tech/ai/.

[45] Zihang Xiang, Tianhao Wang, Wanyu Lin, and Di Wang. Practical differentially private and byzantine-resilient federated learning. *Proceedings of the ACM on Management of Data*, 1(2):1–26, 2023.

[46] Zihang Xiang, Tianhao Wang, and Di Wang. Preserving node-level privacy in graph neural networks. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4714–4732. IEEE, 2024.

[47] Hanshen Xiao, Zihang Xiang, Di Wang, and Srinivas Devadas. A theory to instruct differentially-private learning via clipping bias reduction. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2170–2189. IEEE Computer Society, 2023.

[48] Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Ahmed Salem, Victor Rühle, Andrew Paverd, Mohammad Naseri, Boris Köpf, and Daniel Jones. Bayesian estimation of differential privacy. In *International Conference on Machine Learning*, pages 40624–40636. PMLR, 2023.

[49] Danfeng Zhang and Daniel Kifer. Lightdp: Towards automating differential privacy proofs. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, pages 888–901, 2017.

[50] Jun Zhang, Xiaokui Xiao, and Xing Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *Proceedings of the 2016 international conference on management of data*, pages 155–170, 2016.

[51] Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. In *International Conference on Artificial Intelligence and Statistics*, pages 4782–4817. PMLR, 2022.

A Content for reference

A.1 Neyman–Pearson Lemma

Theorem 6 (Neyman–Pearson lemma [32])**.** *Let P and Q be probability distributions on Ω with densities p and q, respectively. Define $L(x) = \frac{p(x)}{q(x)}$. For hypothesis testing problem*

$$\mathbf{H_0}: P, \quad \mathbf{H_1}: Q$$

For a constant $c > 0$, suppose that the likelihood ratio test which rejects $\mathbf{H_0}$ when $L(x) \le c$ has $\mathrm{FP} = a$ and $\mathrm{FN} = b$, then for any other test of $\mathbf{H_0}$ with $\mathrm{FP} \le a$, the achievable false negative rate is at least b.

Neyman–Pearson lemma says that the most powerful test (optimal false negative rate) at fixed false positive rate is the likelihood ratio test.

A.2 Hoeffding's Inequality

Theorem 7 ([43])**.** *Suppose $\bar{X} = \frac{1}{n}\sum_i X_i$. where $a \le X_i \le b$ are independent, Then for any $t > 0$,*

$$\Pr[|\bar{X} - \mu| \ge t] \le 2\exp\left(-\frac{2nt^2}{(b-a)^2}\right)$$

Note that X_i does not necessarily need to be identically distributed. In our case, $a = 0, b = 1, \mu = p_f^e$, hence, setting

$$2\exp\left(-2nv^2\right) = 2\cdot(1-\gamma)$$

give us $v = \sqrt{\frac{1}{2n}\log\frac{1}{1-\gamma}}$.

B Proofs

B.1 Proof of Property 1

Proof. By fixing $B_{-i} = b_{-i}$, $\mathcal{M}$ is f-DP means that message distribution $M|B_i = 0, B_{-i} = b_{-i} \overset{f-\mathrm{DP}}{\sim} M|_{B_i=1,B_{-i}=b_{-i}}$, because M is only formed based on $\mathcal{M}$'s output and X_i is only included into exactly one run of $\mathcal{M}$.

By post-processing property of f-DP, we have

$$\hat{B}_i|_{B_i=0,B_{-i}=b_{-i}} \overset{f-\mathrm{DP}}{\sim} \hat{B}_i|_{B_i=1,B_{-i}=b_{-i}}$$

as $\hat{B}_i$ is post-processing of m. □

B.2 Proof for Lemma 1

Proof. Denote $S = \sum_{i=1}^n c_i P_i$ and $S' = \sum_{i=1}^n c_i P_i'$ as our null and alternative hypothesis. Consider an arbitrary decision rule $\mathcal{R}$ that takes a sample from S or S' and rejects the null. And suppose we have the false positive rate $\alpha_{\mathcal{R}} = \mathbb{E}_S[\mathcal{R}]$, then

$$\alpha_{\mathcal{R}} = \int_{\{\mathcal{R}(x)=1\}} \sum_{i=1}^n c_i P_i(x)\mathrm{d}x = \sum_{i=1}^n c_i \int_{\{\mathcal{R}(x)=1\}} P_i(x)\mathrm{d}x$$

Let $\alpha_i = \int_{\{\mathcal{R}(x)=1\}} P_i(x)\mathrm{d}x$, which is the false positive rate achieved by rule $\mathcal{R}$ for distinguishing P_i V.S. P', then $\alpha_{\mathcal{R}} = \sum_{i=1}^n c_i\alpha_i$. The false negative rate of $\mathcal{R}$ distinguishing S V.S. S' is $\beta_{\mathcal{R}} = 1 - \mathbb{E}_{S'}[\mathcal{R}]$. We have that

$$\begin{aligned}
1-\beta_{\mathcal{R}} =& \mathbb{E}_{S'}[\mathcal{R}] = \int_{\{\mathcal{R}(x)=1\}} \sum_{i=1}^n c_i P_i'(x)\mathrm{d}x \\
=& \sum_{i=1}^n c_i \int_{\{\mathcal{R}(x)=1\}} P_i'(x)\mathrm{d}x \\
\overset{A}{\le}& 1 - \sum_{i=1}^n c_i f(\alpha_i) \\
\overset{B}{\le}& 1 - f(\sum_{i=1}^n c_i\alpha_i) = 1 - f(\alpha_{\mathcal{R}}).
\end{aligned}$$

This means that $\beta_{\mathcal{R}} \ge f(\alpha_{\mathcal{R}})$, which means that $S \overset{f-\mathrm{DP}}{\sim} S'$ by definition. A is because $P_i \overset{f-\mathrm{DP}}{\sim} P_i', \forall i = 1, \cdots, n$. B is because Jensen's inequality and trade-off function is convex. □

B.3 Proof for Corollary 1

Proof. According to Property 1, if $\mathcal{M}$ is f-DP, we have $\forall i \in [n], b_{-i} \in \{0,1\}^{n-1}$,

$$\hat{B}_i|_{B_i=0,B_{-i}=b_{-i}} \overset{f\text{-DP}}{\sim} \hat{B}_i|_{B_i=1,B_{-i}=b_{-i}},$$

For the conditional distribution $\hat{B}_i|_{B_i=0}$ It is equal to the following distribution in convex combination form

$$\sum_{b_{-i}\in\{0,1\}^{n-1}} \Pr[B_{-i}=b_{-i}]\,\hat{B}_i|_{B_i=0,B_{-i}=b_{-i}}$$

the same also applies to $\hat{B}_i|_{B_i=1}$. By Lemma 1, we have

$$\hat{B}_i|_{B_i=0} \overset{f\text{-DP}}{\sim} \hat{B}_i|_{B_i=1}.$$

□

B.4 Proof for Theorem 2

Proof. Define a hypothesis testing problem as follows.

$$\mathbf{H_0}: B_i = 0, \quad \mathbf{H_1}: B_i = 1.$$

I.e., $\hat{B}_i$ the result of our hypothesis testing. For any decision rule $\mathcal{R}$, leading to false positive rate $\alpha_{\mathcal{R}}$, as governed by the trade-off function, we must obtain false negative rate $\beta_{\mathcal{R}} \geq f(\alpha_{\mathcal{R}})$. Because $\hat{B}_i|B_i=0 \overset{f\text{-DP}}{\sim} \hat{B}_i|B_i=1$.

Based on the above decision rule $\mathcal{R}$, we expand the mutual information quantity as follows. (H is the entropy function and h is the binary entropy function).

$$\begin{aligned}
\mathrm{MI}(G;\hat{G}) =& \mathrm{H}(\hat{G}) - \mathrm{H}(\hat{G}|G) \\
=& h(\Pr(\hat{G}=0)) - p\cdot \mathrm{h}(\Pr(\hat{G}=0|_{G=1})) \\
& - (1-p)\cdot \mathrm{h}(\Pr(\hat{G}=0|_{G=0})) \\
=& \mathrm{h}(p\beta_{\mathcal{R}} + (1-p)(1-\alpha_{\mathcal{R}})) - p\cdot \mathrm{h}(\beta_{\mathcal{R}}) \\
& - (1-p)\cdot \mathrm{h}(1-\alpha_{\mathcal{R}}) \\
\overset{\triangle}{=}& F(\alpha_{\mathcal{R}},\beta_{\mathcal{R}},p)
\end{aligned} \tag{28}$$

with tedious calculation, we have

$$\frac{\partial F}{\partial \beta_{\mathcal{R}}} = p\log\frac{\beta_{\mathcal{R}} - \beta_{\mathcal{R}} t}{t - \beta_{\mathcal{R}} t} \tag{29}$$

where $t = p\beta_{\mathcal{R}} + (1-p)(1-\alpha_{\mathcal{R}})$, we want to show that $\frac{\partial F}{\partial \beta_{\mathcal{R}}} \leq 0, \forall \alpha_{\mathcal{R}}, \beta_{\mathcal{R}}$ governed by the trade-off function. We only need to show that $\frac{\beta_{\mathcal{R}} - \beta_{\mathcal{R}} t}{t - \beta_{\mathcal{R}} t} \leq 1$. Expand this inequality, all boils down to check if $(1-p)(1-\alpha_{\mathcal{R}} - \beta_{\mathcal{R}}) \geq 0$, which is true because $\alpha_{\mathcal{R}} + \beta_{\mathcal{R}} \leq 1$ as governed by the trade-off function.

Hence, we have

$$\begin{aligned}
F(\alpha_{\mathcal{R}},\beta_{\mathcal{R}},p) \leq& F(\alpha_{\mathcal{R}}, f(\alpha_{\mathcal{R}}), p) \\
\leq& \max_{x\in[0,1]} F(x, f(x), p) \\
\overset{def}{=}& \max_{x\in[0,1]} F_f(x,p)
\end{aligned}$$

which is our result in Theorem 2. □

B.5 Proof for Theorem 3

Proof. We follow the setups in proof for Theorem 2 in Appendix B.4.

$$\begin{aligned}
\mathrm{MI}(B_i;\hat{B}_i) =& \mathrm{H}(B_i) - \mathrm{H}(B_i|\hat{B}_i) \\
=& h(\frac{1}{2}) - \mathrm{H}(E_i|\hat{B}_i) \\
=& 1 - \mathrm{H}(E_i|\hat{B}_i) \\
\geq& 1 - \mathrm{H}(E_i) \\
=& 1 - \mathrm{h}(p_i^e)
\end{aligned} \tag{30}$$

As conditioning reduces entropy. Combining the fact that $\mathrm{MI}(B_i;\hat{B}_i) \leq u_f(\frac{1}{2})$ due to Theorem 2, we get the result we want in Theorem 3.

□

B.6 Proof for Theorem 4

Proof. For independent Bernoulli random variables $X_1 \sim \textbf{Bernoulli}(a)$ and $Y_1 \sim \textbf{Bernoulli}(b)$, if $a \geq b$, we have

$$\Pr[X_1 \geq t] \geq \Pr[Y_1 \geq t], \forall t \in \mathbb{R} \tag{31}$$

for random variable X_i, Y_i satisfying Equation (31), we call X_i stochastically dominates Y_i.

For all $t \in \mathbb{R}$, if X_i stochastically dominates Y_i for $i = 1,2$, we have

$$\begin{aligned}
\Pr[X_1 + X_2 \geq t] =& \mathbb{E}_{X_1}\left[\Pr_{X_2}[X_2 \geq t - X_1]\right] \\
\geq& \mathbb{E}_{X_1}\left[\Pr_{Y_2}[Y_2 \geq t - X_1]\right] \\
=& \mathbb{E}_{Y_2}\left[\Pr_{X_1}[X_1 \geq t - Y_2]\right] \\
\geq& \mathbb{E}_{Y_2}\left[\Pr_{Y_1}[Y_1 \geq t - Y_2]\right] \\
=& \Pr[Y_1 + Y_2 \geq t]
\end{aligned} \tag{32}$$

i.e., $X_1 + X_2$ also stochastically dominates $Y_1 + Y_2$, by induction, we have $\Pr[\sum_i X_i \geq t] \geq \Pr[\sum Y_i \geq t], \forall t \in \mathbb{R}$ if X_i stochastically dominates Y_i for $i \in [n]$ and all X_i, Y_i are mutually independent. As $p_i^e > p_f^e, \forall i \in [n]$, setting $X_i = E_i$ and $Y_i = S_i$ $\forall i \in [n]$ gives us the result in Theorem 4.

□

B.7 Proof for Theorem 5

Proof. The first inequality is already proven in Theorem 3, the last inequality is because $\mathrm{MI}(B_i;\hat{B}_i|B_{-i}) = \mathbb{E}_{B_{-i}}\left[\mathrm{MI}(B_i;\hat{B}_i)\right] \leq u_f(\frac{1}{2})$, as $\mathrm{MI}(B_i;\hat{B}_i) \leq u_f(\frac{1}{2})$ by theorem 2.

For random variable X, Y, Z, using the chain rule of mutual information, we have

$$\begin{aligned}\mathrm{MI}(X;Y,Z) =& \mathrm{MI}(X;Y|Z) + \mathrm{MI}(X;Z) \\ =& \mathrm{MI}(X;Z,Y) \\ =& \mathrm{MI}(X;Z|Y) + \mathrm{MI}(X;Y)\end{aligned} \tag{33}$$

if we have X is independent Z (which means $\mathrm{MI}(X;Z) = 0$), the above equation give us $\mathrm{MI}(X;Y|Z) \geq \mathrm{MI}(X;Y)$. Setting $B_i = X, \hat{B}_i = Y, B_{-i} = Z$ give us the result for the model inequality of Equation (25). □