Eureka: A General Framework for Black-box Differential Privacy Estimators

Yun Lu
University of Victoria
Victoria, Canada
yunlu@uvic.ca

Malik Magdon-Ismail
Rensselaer Polytechnic Institute
Troy, USA
magdon@cs.rpi.edu

Yu Wei
Purdue University
West Lafayette, USA
yuwei@purdue.edu

Vassilis Zikas
Purdue University
West Lafayette, USA
vzikas@cs.purdue.edu

ABSTRACT

Differential Privacy (DP) is a gold standard of privacy. Nonetheless, one challenge for non-privacy experts to utilize DP, is the difficulty in analyzing the privacy of their often quite complex tasks.

Our work aims to address the above limitation. In a nutshell we devise a methodology for domain experts with limited knowledge of security to estimate the (differential) privacy of an *arbitrary* mechanism. Our Eureka moment is the utilization of a link—which we prove—between the problems of DP parameter-estimation and Bayes optimal classifiers in machine learning, which we believe can be of independent interest. Our estimator methodology uses this link to achieve two desirable properties: (1) it is *black-box*, i.e., does not require knowledge of the underlying mechanism, and (2) it has a theoretically-proven accuracy, which depends on the underlying classifier used. This allows domain experts to design mechanisms that they conjecture offer certain (differential) privacy guarantees—but maybe cannot prove it—and apply our method to confirm (or disprove) their conjecture.

More concretely, we first prove a new impossibility result, stating that for the classical DP notion there is no black-box poly-time estimator of (ε, δ)-DP. This motivates a natural relaxation of DP, which we term *relative DP*. Relative DP preserves the desirable properties of DP—composition, robustness to post processing, and robustness to the disclosure of new data—and applies in most practical settings where privacy is desired. We then devise a black-box poly-time (ε, δ)-relative DP estimator—the first to support mechanisms with large output spaces while having tight accuracy bounds. As a result of independent interest, we apply this theory to develop the *first* approximate estimator for the standard, i.e., non-relative, definition of *Distributional Differential Privacy* (DDP) – aka noiseless privacy.

To demonstrate both our theory and its practicality, we devise and benchmark a proof-of-concept implementation of our estimator. In reasonable execution time, our implementation reproduces tight, analytically computed ε, δ trade-off of Laplacian and Gaussian mechanisms—to our knowledge, the first black box estimator to do so, and for the Sparse Vector Technique, our outputs are comparable to that of a more specialized state-of-the-art (ε, δ)-DP estimator.

KEYWORDS

Differential privacy; Bayes classifier; machine learning; kNN

1 INTRODUCTION

As big-data algorithms, e.g., machine learning (in short, ML), become more sophisticated and ubiquitous, the need to ensure privacy for sensitive data becomes ever more prominent. Differential privacy (DP) is one broadly accepted notion of privacy for a wide range of applications. Despite numerous milestone results over decades of research, there is only a handful of DP mechanisms whose privacy can be analytically calculated. Furthermore, these mechanisms can often not be applied to protect the privacy of queries that invoke complex algorithms, such as machine learning on private data. This limits the accessibility of DP to application domain experts who are not trained in security.

Informally, a mechanism $\mathcal{M}$ is (ε, δ)-DP if for any pair of *neighboring* databases D, D', the output distributions of $\mathcal{M}(D)$ and $\mathcal{M}(D')$ are (ε, δ)-close. Parameters ε and δ quantify the DP of $\mathcal{M}$—the smaller the more private. The two parameters address different aspects of privacy: informally, ε quantifies how much the privacy of each individual record is protected, and δ can be seen as the probability that all privacy guarantees are given up. Thus, naturally one is interested in keeping δ tiny. However, in most applications—and for most mechanisms—there is an inherent trade-off between these two parameters, i.e., aiming for tiny δ comes at the cost of high values of ε. Charting this trade-off is thus important in deciding if a mechanism is a good fit for a given application.

In this work we define the *privacy spectrum*, also referred to as *DP-spectrum*, of a mechanism $\mathcal{M}$, denoted as $\delta_{\mathcal{M}}(\varepsilon)$,[1] to be the optimal (i.e., minimum) δ achievable for a given ε. We then devise a methodology for estimating the privacy spectrum of any given $\mathcal{M}$, while using $\mathcal{M}$ in a black-box manner. Our methodology uses ML theory to provide provable accuracy guarantees, a common approach when designing efficient mechanisms in ML and cryptography. We prove that our ML-based method estimates the optimal δ within a sufficiently small error, which diminishes with the number of samples (runtime) that the estimator uses. We then empirically demonstrate that the asymptotically-predicted behavior kicks-in already for a small number of samples. In the following we outline our main results:

Our first result is on the impossibility of a poly-time black-box (ε, δ)-DP estimator: there is no poly-time black-box estimator to compute the DP-spectrum of an arbitrary input mechanism (see

[1] To keep our notation simple, whenever the mechanism $\mathcal{M}$ is clear from the context we drop it from the notation of the DP-spectrum, i.e., we write $\delta(\varepsilon)$ instead of $\delta_{\mathcal{M}}(\varepsilon)$.

Theorem 4.5). This result justifies a recent line of work [1–8] that takes aim at the challenge of black-box DP estimator by proposing methods to *empirically estimate* the DP-spectrum of a mechanism. The desirable properties of such estimators are: *accuracy, generality, and efficiency*.

Accuracy requires that the estimated DP-spectrum for the mechanism $\mathcal{M}$ should be close to true DP-spectrum of $\mathcal{M}$. There are two modes in which one can empirically analyze the DP spectrum of a mechanism.

(1) *Verify* if a mechanism satisfies a given (ε, δ)-DP requirement. Typically the approach is to estimate a *lower* (*upper*) *bound* on the DP parameter(s) [5–7] and use these bounds to decide if the privacy is violated. The bounds produced can be loose, and so the outcome of the verification is not always conclusive.
(2) A stronger and more useful statement is to estimate the full DP-spectrum of the mechanism, by producing *tight (upper and lower) bounds* on the privacy parameters. This is the task we tackle in this work. To our knowledge the only other work which attempted such a tight estimation is ADP-Estimator [8] which however can only be used for mechanisms with a small output domain. (We refer to Section 2 for a detailed comparison.)

A thread in the aforementioned prior work takes a heuristic approach, offering primarily empirical estimates of the privacy parameters. In contrast, in this work we develop a framework that allows for theoretical guarantees on the estimated privacy of a (DP) mechanism. Our methodology can in principle be applied to estimate the privacy obtained by arbitrarily complex mechanism, as it uses this mechanism in a black-box manner. Importantly, we validate the theory and demonstrate the potential of our method to yield a practical estimator for various tasks, via a proof-of-concept implementation of our estimator. Concretely, in order to demonstrate the accuracy of our theory and the potential practicality of our estimator, we benchmark it against mechanisms whose theoretical properties are already well understood, like the Laplacian and Gaussian, as well as those with varying implementations, like the Sparse Vector Technique (SVT).

Generality mandates that the estimator should work for *any* mechanism. One way to achieve this is by making the estimator agnostic as to what the mechanism does, i.e., the mechanism is used in a *black-box* manner. Such a black-box estimator, which is the type we develop, only interacts with the mechanism in an input/output manner. In contrast, a *white-box* (aka non-black-box) estimator needs to know the (pseudo-code) of the mechanism whose privacy is to be estimated. An orthogonal feature of estimators regarding generality is whether they estimate only the ε parameter (aiming for the less flexible ε-DP) or, as we do in this work, estimate the full DP-spectrum which quantifies the ε-δ trade off. The latter is more general, as ε-DP is the same as $(\varepsilon, 0)$-DP (setting $\delta = 0$).

Efficiency is necessary for an estimator to be useful in practice. As we discuss in Section 2, depending on the actual size of the datasets and, more intriguingly, the output space of the mechanism whose privacy is being estimated, certain methodologies that exhaustively process the output space, such as [5, 6], quickly become impractical, especially for large output spaces. In fact, to our knowledge, ours is the first tight, black-box, and theory-backed (ε, δ)-DP estimator that can handle even mechanisms with a large (and even uncountable) output space. (We offer more comprehensive comparison of our estimator with existing methods in Section 2, *cf* Table 1.)

1.1 Our Contributions

We put forth a general framework for constructing and analysing black-box DP estimators, and propose, analyze, and benchmark a concrete instantiation. At a high level, the main insight driving our results is that the task of a black-box DP-spectrum estimator can be re-cast as a specially-crafted classification problem, which can then be analyzed and solved by ML techniques. In particular, given a data set and a (black-box) mechanism, we devise a new classification task whose optimal classifier can be directly linked to the DP-spectrum of the mechanism. Thus we can employ tools from the the literature of this optimal classifier to estimate (theoretically and empirically) the DP-spectrum of the given mechanism. Concretely, using tools from statistical learning theory, we are able to obtain tight bounds on the performance of this optimal classifier, which leads to our estimator for the DP-spectrum of the black-box mechanism. In the following we elaborate on some of the main points and techniques, and give pointers to the paper sections that include the detailed treatment.

Relative Differential Privacy (Section 4) First, we ask if it is even possible to efficiently and exactly estimate the (ε, δ)-DP-spectrum of an arbitrary mechanism. The answer is that *no* efficient black-box DP estimator can exactly compute the ε-δ privacy trade off.

A straw-man attempt to circumvent the above would be to relax the "exactness" requirement and aim for an approximate estimator. Nonetheless, we show that even if we relax the exactness in a very generous manner to allow both for error probability and for an approximation factor, the above impossibility cannot be circumvented. To this direction, we devise the relaxation of *randomized approximate* estimator—i.e., one that with high probability, $1-\beta$, approximates the DP parameters of the mechanism up to an (additive) approximation factor α. Unfortunately, as we show in Theorem 4.5, this relaxation is of little use, as an estimator is also in general impossible for reasonable parameters α and β.

A second attempt to circumvent the above impossibility could be to settle for one of the relaxed definitions of DP from the literature, such as Renyi DP [9] and (Zero-)Concentrated DP [10, 11]. Unfortunately, they also do not accept efficient black-box estimators. In fact, one can verify that the proof idea of our impossibility theorem (Thm. 4.5) applies also to these relaxations.

Motivated by the above, here, we introduce a natural relaxation of (ε, δ)-DP, which we term *relative differential privacy* (relative DP for short) (Sec. 4.1), that circumvents the impossibility (and for which, as we show, an approximate estimator is indeed possible.) Informally, an $(\varepsilon, \delta, \mathcal{T})$-relative DP mechanism is one which satisfies (ε, δ)-DP for databases in a given set $\mathcal{T}$. We believe that such a relaxation is well justified by the key uses of DP in practice: Typically there are limited datasets that one might have access to, so requiring DP to apply for any dataset might be overreaching when it comes to estimating privacy in real-world applications.

In fact, we prove that relative DP has many of the desirable properties of DP that make it useful in a wide range of applications,

more prominently its "future-proofness". In a nutshell, we prove in a sequence of results (Proposition 4.8-4.11) that relative DP is reasonably robust to adding new databases to the set $\mathcal{T}$—informally, the privacy of the estimated mechanism is never worse than the privacy of the mechanism on the new set $\mathcal{T}$. Subsequently, we prove that relative DP preserves the common desirable notions of DP, namely sequential/parallel composition, and robustness to post-processing.

(Relative) DP Estimator (see Section 5). Armed with the notion of relative DP, we then proceed to the task of devising and analysing a relative DP(-spectrum) estimator, by linking DP to an optimal (i.e., Bayes) classifier for a carefully constructed classification problem that uses the mechanism as a black-box, and the databases in $\mathcal{T}$. Because we are after a method with theory-backed guarantees, we focus on the well studied k-Nearest-Neighbor (kNN) algorithm [12]; nonetheless, our methods can be instantiated (in a mostly plug-and-play manner) by any other classification algorithm to achieve similar guarantees but different performance.

To help the reader build intuition on the basic principles of our methodology, we start with the simplest instance of relative DP, where the set $\mathcal{T}$ includes just a single database; we show how to estimate the privacy of any single given record (i.e., for a specific pair of neighboring databases). We stress that due to this setting's (over)-simplified nature, results in this setting are of-course not particularly relevant for assessing the privacy of the given mechanism. Nonetheless, we believe it offers a smoother way to ease into the ideas of our description and analysis of our general estimator. The actual result is then derived by removing the above simplification.

In more detail, focusing on the above (oversimplified) setting, we start by presenting a general method to convert the *risk* (or error) of a Bayes/optimal classifier to the δ privacy parameter of a DP mechanism (Theorem 5.4). Then, in Lemma 5.6, we convert the convergence theorem[2] of a classifier to tight bounds on the accuracy of our relative DP estimator. We apply this lemma to the kNN classifier in Theorem 5.7. The final step to construct our (relative) DP estimator is to extend the set $\mathcal{T}$ to be any polynomial-size set of databases. The idea is to employ the above singleton-$\mathcal{T}$ algorithms for each of the databases in $\mathcal{T}$ and then use Proposition 4.8 to bound the parameters with respect to the whole set $\mathcal{T}$. Our main results are the Algorithms in Figs. 1 and 2 for estimating the relative DP-spectrum, and the accompanying Theorem 5.10 which proves its convergence rate to the true relative DP-spectrum.

Distributional Differential Privacy (see Section 6) At the heart of the nonexistence of an (even approximate) estimator for DP that we proved is the standard problem in ML classification: The input distribution of the algorithm whose parameters we are trying to estimate is completely unknown, and in the worst case, learning it would require an infeasible number of (or even infinitely many) samples. In fact, knowledge of the data distribution can be used to replace the "relative" (to a specific $\mathcal{T}$) restriction of our treatment. This makes our framework directly applicable to "noiseless" versions of DP such as the well known *Distributional Differential Privacy* (DDP) notion [13]. In a nutshell, these notions propose taking advantage of the inherent entropy that is included in common datasets to reduce the amount of noise needed to achieve the closeness metric of DP (see Section 3.2 for an overview.) We show that, under the assumption of independently distributed database rows, our relative DP estimator framework can be employed to estimate the DDP parameters of a mechanism. To our knowledge, this yields the *first black-box DDP estimator*. We believe that both the general paradigm and the estimator itself are of independent interest to the ML/AI research, where the question of whether a given algorithm achieves any meaningful notion of (noiseless) privacy has been circulating for a long time.

Validating our Theory & Benchmarking our Estimator (see Section 7). To complement the theory, we validate our (asymptotic) bounds empirically. Our experiments demonstrate that with a moderate number of samples, we can already showcase the concrete practicality of our estimator. It is worth noting that in a milestone result on ML theory, Antos *et al.* [14] proved that there is no fixed finite sample-size beyond which one can universally bound the convergence rate of a Bayes risk estimator. This makes our empirical validation the ideal, if not only way to validate our theory and demonstrate that the asymptotic predictions kick in for moderately-sized samples. (Such an empirical validation of asymptotic theory is common in both the cryptography/privacy and in the ML literature). The findings of our empirical analysis are described next.

By testing existing DP mechanisms whose privacy can be computed exactly using analytical methods, we demonstrate a (nearly exact) match of these analytical values and the output of our estimator. Theorem 5.7 states the relationship between error and number of samples—e.g. doubling number of samples decreases error by $1/\sqrt{2}$. This relationship allows users who run our estimator to calculate the number of samples required for a theoretically guaranteed desired error. In practice, far fewer samples are needed. Our algorithm runs in $O(mn)$, where m is the number of neighboring databases tested (this is a necessary dependency to estimate any mechanism, since a mechanism's behavior on different databases can vary drastically) and n is the number of samples. Furthermore, achieving cryptographically small error in δ is feasible: our evaluation with $\delta = 10^{-5}$ error needs just 2^{26} samples which takes 10 minutes using a simple textbook implementation of kNN. In addition, we provide experiments for SVT, a popular mechanism with various (sometimes incorrect) implementations. Our privacy estimates are comparable to the state-of-the-art estimates which use a specialized algorithm aimed towards mechanisms with limited output space (as it iterates over this space) [8].

The combination of theoretically and concretely tight accuracy bounds means our estimator can reveal the full privacy spectrum of a mechanism. By quantifying the ε, δ privacy parameter trade-off (under a set of databases), we can not only verify the correctness of a mechanism's implementation, but also compare the privacy of two different mechanisms.

2 RELATED WORK

Below, we discuss previous work on privacy estimators, categorizing them by their method.

[2] A convergence theorem describes the difference between the accuracy of a classifier (such as kNN), and the accuracy of the theoretical optimal classifier.

	Access to $\mathcal{M}$	$\mathcal{M}$ with large output space	Accuracy	Methods
StatDP [5]	Semi-black-box	No	Lower bounds	Hypothesis testing
DP-Finder [6]	White-box	No	Lower bounds	Sampling and optimzation
DP-Sniper [7]	Black-box	Yes	Lower bounds	Classifier
DPL [15]	Black-box	Yes	Lower bounds	Kernel Density estimator
ADP-Estimator [8]	Black-box	No	Upper and lower bounds	Distribution estimator
Our Work	Black-box	Yes	Upper and lower bounds	Classifier (e.g., kNN)

Table 1: Summary of comparisons between our work and previous works.

Programming Language-based methods. This line of works [1–4] uses language-based methods to automatically verify whether or not a mechanism satisfies certain level of differential privacy. These methods require *white-box* access to the tested mechanism—such as access to the tested mechanism's code, even requiring manual annotations on the code. They are particularly useful in formally verifying if the implementation of some known mechanisms is correct or buggy. In particular, these estimators automatically search and infer proof of the DP property for the tested mechanism, hence the result (satisfying DP or not) can be very accurate if they do succeed. However, automated verification may sometimes fail to complete its task to verify the mechanism's DP parameters. For example, [4] reports that LightDP [1] is unable to disprove faulty variants of PrivTree [16], because the variants have a probabilistic main loop that terminates eventually but its number of iterations can't be bounded. The main advantage of our work compared to this line of works is that we pursue a probabilistic, data-driven, and black-box approach, and thus can be applied to general mechanisms, even proprietary software or heuristic attempts by ML researchers, without access to the mechanism's code.

Probabilistic testing methods. This line of works [5–8, 15] uses statistical tools and is based on sampling the mechanism's inputs/outputs. Specifically, the works [5–7, 15] focused on the task of lower-bounding the DP parameter of a mechanism—that is, asserting that the tested mechanism cannot achieve (beyond a) certain level of differential privacy. The core challenge then is to find a witness of the DP violation for privacy parameters beyond this level. StatDP [5] requires semi-black-box access to the tested mechanism, as one of its post-processing requires running the tested mechanism on input data without any noise. DP-Finder [6] requires the tested mechanism's algorithm (which it relies on white-box access to) to be differentiable, so that excludes common operations such as arbitrary loops or hash functions. This requirement considerably limits the class of mechanisms the method applies to, and excludes common differential private techniques such as SVT [17] and Randomized Response [18]. DP-Sniper [7] and the most recent work DPL [15] use the black-box approach and are designed for general mechanisms. DPL [15] improves upon DP-Sniper [7] by avoiding the process of "event selection"—a major obstacle to finding privacy violation witness. This is achieved via a method called kernel density estimation. However, similar to all the above works in this thread, DP-Sniper and DPL aim to test the ε-DP property, and constructs algorithms that find only a *lower bound* of the privacy parameter ε for the tested mechanism on neighboring databases. In comparison, the main goal of our work is to provide a tight characterization (i.e., *both* upper and lower bounds) on both the ε and δ privacy parameters.

ADP-Estimator [8] aims to test the (ε, δ)-DP property for a mechanism, and discuss the relationship between the accuracy in estimated privacy parameters and the number of samples required. While the goals of our work align with that of [8], our approach is vastly different. ADP-Estimator presents one specific method of empirically estimating the mechanism's output distributions for a single pair of neighboring databases. In comparison, we develop a general framework that gives a formal treatment of the DP parameter-estimation problem and links it to the rich ML theory on classification algorithms, hence our method can derive a family of privacy estimators by using different classifiers in a plug-and-play manner. In addition, the ADP-Estimator [8] is limited: by enumerating the tested mechanism's output space, their algorithm requires this space to be a finite (and small) set. In contrast, our estimator instantiation using kNN classifier does not have such limitations. As further evidence of our method's advantage, we estimate the Gaussian mechanism (Section 7), which hadn't been reported by either DP-Sniper and DPL (they only aim to ε-DP) or ADP-estimator (it is inefficient to test mechanism with large output space).

Machine Learning for DP The connection between machine learning and DP estimation has recently attracted attention in the ML/AI literature. A recent line of works [19–22] investigated a connection between DP and empirically estimatable statistical distance. In a nutshell, the goal of these works is to bound the distinguishing advantage between distributions $\mathcal{M}(D)$ and $\mathcal{M}(D')$ (which directly relates to their statistical distance) for a DP mechanism $\mathcal{M}$ and a pair of (neighboring) database D and D'. Specifically, given a (ε, δ)-DP mechanism, these works upper bound the statistical distance between $\mathcal{M}(D)$ and $\mathcal{M}(D')$. This in turn implies a lower bound on δ as a function of ε and the statistical distance between $\mathcal{M}(D)$ and $\mathcal{M}(D')$. In contrast, our results use a pair of carefully crafted distributions (not $M(D)$ and $M(D')$) which allows us build an exact link between the DP-spectrum and a Bayes optimal risk. By then estimating this risk nonparametrically, we are able to get tight statistical upper and lower bounds on the achievable δ parameter for every given ε, hence giving an accurate characterization of the entire DP-spectrum. Devising and analyzing these new distributions—and the connection to the DP-spectrum—is a key novelty here and can be seen as a non-trivial extension of Le Cam's (lower-only) bound [23]: we present equality rather than just a lower bound, which we can use to tightly bound (both upper and lower) the accuracy of our estimate of δ for every given ε.

Lastly, Gilbert and McMillan [24] discuss the lower bound of the sample complexity of verifying whether some specific (ε, δ)-DP is satisfied. Their work is useful to answer what type of privacy parameter verification task is feasible. In contrast, our work devises a concrete method of *tightly estimating* (relative) differential privacy. To achieve this, we also develop sample complexity results which are orthogonal to [24].

3 PRELIMINARIES

We introduce the privacy definitions for which we will construct our privacy estimators. Moreover, we introduce relevant background on classifiers, in particular the kNN classifier.

3.1 Differential Privacy

Informally, *differential privacy* (in short, DP) [25] is defined via an experiment between a query party P and a *curator* C, who has access to a database D. P wishes to make a query Q on the database, and C wants to answer this query in a way that protects the privacy of any individual record. This property is achieved by C using a randomized algorithm, aka *mechanism*, to answer P's queries, in a way that does not destroy accuracy—i.e., the outcome of the mechanism is not too far from the true answer to the query—while respecting the privacy of any individual record $X \in D$— i.e., P (or in fact any P' with arbitrary side-information on the database) has only a small chance in telling weather or not X was used in answering the query. To make this formal, we state here the definition of DP (cf. [26] for an excellent treatment of DP and its properties.)

*Definition 3.1 (**Mechanism**).* Let $\mathcal{U}$ be the set of all possible database records. Let $\mathcal{X} = \mathcal{U}^*$ be the set of all databases where each database row is from $\mathcal{U}$. Let $\mathcal{O}$ be the set of all possible output strings. Then a mechanism $\mathcal{M} := \mathcal{X} \mapsto \mathcal{O}$ is a (randomized) algorithm that takes as input a database from the input space $\mathcal{X}$, and produces an output from the output space $\mathcal{O}$.

In DP, we are interested in whether our mechanism reveals information on individual database records. Thus, we consider the output of our mechanism on pairs of databases called *neighbors*, where one neighbor contains a particular individual record, and the other does not.

*Definition 3.2 (**Neighboring Databases**).* A pair of databases $D, D' \in \mathcal{X}$ is *neighboring*, denoted $D \simeq D'$ if D' can be obtained from D by removing one row.

A mechanism is DP if its output given a database is similar to its output given the database's neighbor.

*Definition 3.3 (**Differential Privacy (DP)** [25]).* A mechanism $\mathcal{M} := \mathcal{X} \mapsto \mathcal{O}$ is (ε, δ)-differentially private if for all subset $\mathcal{S} \subseteq \mathcal{O}$ and for all neighboring databases $D \simeq D'$:

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

and

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

where the probability space is over the coin flips of the mechanism $\mathcal{M}$. If $\delta = 0$, we say that $\mathcal{M}$ is ε-differentially private.

3.2 Distributional Differential Privacy (DDP)

The above DP definition is broadly used, but might be inapplicable in cases where utility degrades rapidly even with small noise, such as machine learning with deep networks, whose performance is sensitive to noise in the data. *Distributional differential privacy (DDP)* [13].[3] was suggested as an alternative to DP that can treat such cases. The idea here is that we might often be willing to make an assumption about the entropy (inherent randomness) of the database; in this case, we might be able to avoid using (too much) extra randomness/noise in the mechanism, and instead, rely on this internal randomness of the data to achieve similar privacy guarantees as DP with less to no hit on the output's accuracy. More concretely, in DDP, instead of considering fixed databases D, we consider databases as random variables (r.v.'s) from a distribution π. We denote by D_{-i} as the random variable that is the same as database D, but without its ith row. Denote by D_i the ith row of D. We denote by $\mathsf{Supp}(\cdot)$ as the support of a random variable. Informally, a mechanism $\mathcal{M}$ is DDP for some distribution π and auxiliary information z if its output on some database (r.v.) can be approximated by a function h *without* being given the ith row of this database.

Definition 3.4 (Distributional differential privacy (DDP) [13]). A mechanism $\mathcal{M}$ is $(\varepsilon, \delta, \Delta)$-*distributional differentially private* if there is a function h[4] such that for all $(\pi, Z) \in \Delta$, $D \sim \pi$, for all i, $(x, z) \in \mathsf{Supp}(D_i, Z)$, and all sets $\mathcal{S} \subseteq \mathsf{Range}(\mathcal{M})$,

$$\Pr_{D\sim\pi}(\mathcal{M}(D) \in \mathcal{S} | D_i = x, Z = z) \leqslant e^{\varepsilon} \Pr_{D\sim\pi}(h(D_{-i}) \in \mathcal{S} | D_i = x, Z = z) + \delta,$$

and

$$\Pr_{D\sim\pi}(h(D_{-i}) \in \mathcal{S} | D_i = x, Z = z) \leqslant e^{\varepsilon} \Pr_{D\sim\pi}(\mathcal{M}(D) \in \mathcal{S} | D_i = x, Z = z) + \delta.$$

In the case of distributions π with independently distributed rows, and when $Z = \emptyset$ (there is no auxiliary information), we can greatly simplify the above definition of DDP.

Definition 3.5 (Simplified DDP). Let Δ be a set of distributions on databases where each row is independently distributed. For any $\varepsilon > 0$ and $\delta > 0$, a mechanism $\mathcal{M}$ is $(\varepsilon, \delta, \Delta)$-DDP if for every $\pi \in \Delta$, $i \leqslant n$, $x, x' \in \mathcal{U}$, and $\mathcal{S} \subseteq \mathsf{Range}(\mathcal{M})$, the following inequality holds.

$$\Pr_{D\sim\pi}(\mathcal{M}(D) \in \mathcal{S} | D_i = x) \leqslant e^{\varepsilon} \Pr_{D\sim\pi}(\mathcal{M}(D) \in \mathcal{S} | D_i = x') + \delta,$$

The work of Liu et al. [30] shows that the definition above is equivalent to DDP under the simplifying assumption of independent rows and no auxiliary information, as is common in machine learning.

Lemma 3.6 (Equivalence of definitions [30]). *We denote Def. 3.4 as the* simulation-based *DDP*[5]. *For any $\mathcal{U}$, let Δ be a set*

[3] We focus here on DDP but we believe our approach applies also to alternative type of noisless privacy [27–29].

[4] In [13] h is called the *simulator* in the sense that h "simulates" missing ith row of D, and following notation from a similar concept in security. However, to avoid confusion we simply refer to h as a function.

[5] (1) Following the notation in [13]. (2) Although the lemma in [30] is stated with respect to i.i.d. database rows, an inspection of the proof shows only independence of the rows is required.

of distributions on databases where each row is independent, and $\Delta' = (\Delta, Z = \emptyset)$. *Suppose* $\mathcal{M}$ *is* $(\varepsilon, \delta, \Delta')$*-simulation-based DDP, then* $\mathcal{M}$ *is* $(2\varepsilon, (1+e^{\varepsilon})\delta, \Delta)$*-DDP for our Definition 3.5. Conversely, if* $\mathcal{M}$ *is* $(\varepsilon, \delta, \Delta)$*-DDP for Definition 3.5 then* $\mathcal{M}$ *satisfies* $(\varepsilon, \delta, \Delta')$*-simulation-based DDP.*

3.3 Classification Algorithms

Our treatment uses concepts and results from machine learning (ML) theory to construct our privacy estimator and prove (tight) bounds on its accuracy, i.e., how well it estimates optimal pairs (ε, δ) for the (D)DP definitions. For completeness, here we recall the necessary basic machine learning (ML) background to interpret our results.

Let $\mathcal{O}$ denote the observation space, and let the label (or prediction) space be $\mathcal{Y} = \{0, 1\}$ (e.g., outputting 0 means the classifier predicts the observation is from one distribution and outputting 1 means the classifier predicts the other distribution). Let $\mathcal{P}$ be a joint distribution with the support of $\mathcal{O} \times \mathcal{Y}$, where $\mathcal{O} \times \mathcal{Y} := \{(o, b) : o \in \mathcal{O}, b \in \mathcal{Y}\}$ is a concatenation set. Let $\mathcal{I}(b, y)$ be the *inequality predicate*, i.e., the indicator function outputs 1 if b is not equal to y, otherwise 0.

A *classifier* $h : \mathcal{O} \mapsto \mathcal{Y}$ (also called a *classification algorithm*) is a function from the observation space $\mathcal{O}$ to the prediction space $\mathcal{Y}$. For every observation $o \in \mathcal{O}$, h outputs a bit $b \in \mathcal{Y}$ indicating that h predicts o has label b.

A *risk function* R is defined with respect to a distribution $\mathcal{P}$ on observables—in fact, it is easier to think of $\mathcal{P}$ as a joint distribution of pairs of the type (x, y) where x is an observation and y is its label. R takes a classifier h as input, and computes the probability that a sample drawn from $\mathcal{P}$ is mistakenly classified—i.e., assigned the wrong label—by h; equivalently, R computes the expectation of the above inequality predicate. Formally:

$$R(h) = \Pr_{(x,y)\sim\mathcal{P}}[\mathcal{I}(h(x), y) = 1] = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}}[\mathcal{I}(h(x), y)].$$

We note that in a given application context, the risk $R(h)$ is typically impossible to compute, as the distribution $\mathcal{P}$ is unknown. However, viewing risk $R(h)$ as the expectation of the random variable $\mathcal{I}(h(x), y)$, allows us to derive a good estimator for it: the *testing risk* $\hat{R}_m(h)$ which is defined as the average on a set of independent samples $((x_1, y_1), \cdots, (x_m, y_m)) \sim \mathcal{P}^m$. (We make the sampling process $((x_1, y_1), \cdots, (x_m, y_m)) \sim \mathcal{P}^m$ implicit when it is clear from context). Formally:

$$\hat{R}_m(h) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{I}(h(x_i), y_i),$$

In particular, a well-known result using Hoeffding's inequality allows us to gauge, up to an error probability γ, how close $\hat{R}_m(h)$ is to the true risk $R(h)$:

Theorem 3.7 (**Hoeffding's Inequality** [31]). *With probability* $1 - \gamma$,

$$|\hat{R}_m(h) - R(h)| \leqslant \sqrt{\frac{1}{2m}\ln\frac{2}{\gamma}}.$$

Bayes (optimal) classifiers. A *Bayes (optimal) classifier* h^* with respect to $\mathcal{P}$ is a classifier that has the minimal risk $R(h^*)$ among all the classifiers (with respect to the same $\mathcal{P}$.)

The kNN Classifier. Unfortunately, for the same reason we can not compute R—i.e., because $\mathcal{P}$ is typically unknown[6]—we can also not construct the Bayes classifier h^*. Nonetheless, the ML theory provides us with several "reasonable" classifiers that achieve both good performance, and are close to optimal. One such classifier which is well understood and thoroughly studied in the field of pattern recognition is the *k-Nearest Neighbor (kNN) classifier*—which we use in our paper as a concrete instantiation of our framework. To construct a kNN classifier $h^{\mathsf{NN}}_{k,n}$ with n samples, we simply sample and store n training points $((x_1, y_1), \cdots, (x_n, y_n)) \sim \mathcal{P}^n$. To predict the label of an observation $o \in \mathcal{O}$, $h^{\mathsf{NN}}_{k,n}$ returns the label taking a majority vote of the class labels of its k nearest neighbors (according to the distance function defined on the space) in the stored training points:

$$h^{\mathsf{NN}}_{k,n}(o) = \left\lfloor \frac{1}{k}\sum_{i\in[k]} b_i \right\rceil,$$

where b_i is the label of the i-th nearest neighbor of o, and $\lfloor\cdot\rceil$ is an operator rounding to nearest integer.

The following convergence result for kNN gauges how close the true risk $R(h^{\mathsf{NN}}_{k,n})$ of the kNN classifier $h^{\mathsf{NN}}_{k,n}$ is to the risk of the optimal classifier, $R(h^*)$.

Theorem 3.8 (**Convergence of k-Nearest Neighbor Classifier** [12]). *Let* $\mathcal{P}$ *be a joint distribution with support* $\mathcal{O}\times\mathcal{Y}$*. If the conditional distribution* $\mathcal{P}|\mathcal{Y}$ *has a density,* $\mathcal{O} \subseteq \mathrm{R}^d$*, and* $k = \sqrt{n}$*, then for every* $\alpha > 0$ *there is an* n_0 *such that for* $n > n_0$*,*

$$\Pr[|R(h^{\mathsf{NN}}_{k,n}) - R(h^*)| > \alpha] \leqslant 2e^{-n\alpha^2/(72c_d^2)},$$

where c_d[7] *is the minimal number of cones centered at the origin of angle* $\pi/6$ *that cover* R^d*. Note that if the number of dimensions* d *is constant, then* c_d *is also a constant.*

Notes on using kNN: The astute reader may observe that we require a technical assumption on density when using kNN as our classifier. This is standard assumption in the ML literature and essentially amounts to the observable being smoothly varying. One can easily generalize this to, e.g., also discrete observables because a discrete distribution can be approximated arbitrarily closely by a smooth distribution in one dimension. This means the Bayes optimal risk between the two discrete distributions is arbitrarily close to the Bayes optimal risk between the two arbitrarily close smooth approximations. Furthemore, mechanisms that noise their output via a distribution with density (e.g., Laplace, Gaussian), automatically satisfy the smoothness condition on the density. The reader may also observe that the term c_d in Thm. 3.8 implies our results depend on the dimensionality of the mechanism's output (which we indeed see in Thm 5.7). This 'curse' of output-dimensionality will be inherent to *any* DP estimator due to the direct connection between classifier risk and DP parameters. Nonetheless, many applications exist, such as private counting/range queries and ML data aggregation mechanisms, where output dimension is small. In fact, evidence shows mechanisms with large output dimensionality are typically less accurate (e.g., the private mechanisms [32, 33] for deep learning reduce dimensionality via PCA).

[6] In a typical ML classification experiment, one is able to observe values sampled from $\mathcal{P}$ but does not know the actual distribution.

[7] By Lemma 5.5 of [12], c_d satisfies $c_d \leqslant (1 + 2/\sqrt{2-\sqrt{3}})^d - 1$.

4 RELATIVE DP: MOTIVATION AND DEFINITION

In this section, we will first give an intuitive definition of a perfect and approximate DP estimator. Then, we will motivate *relative DP* with an impossibility result: A black-box poly-time (approximate) estimator for *differential privacy* parameters with tight bounds on accuracy does not exist. Informally, we define a privacy estimator as an algorithm which, given a mechanism $\mathcal{M}$ and an ε value, outputs a δ for which it believes $\mathcal{M}$ is (ε, δ)-DP (symmetrically, it can also be given δ and be asked to estimate ε). An estimator with tight accuracy bounds (α, β) means its output δ will be at most α-far from the optimal solution with probability $1 - \beta$. In other words, it gives a known probability of success, and an upper and lower bound on its output's closeness to the true privacy of the mechanism.

Below, we first define the notion of *optimal* δ given any ε and mechanism $\mathcal{M}$. Note this optimal δ is a point in the DP-spectrum discussed in the introduction. We also define the quantity $\delta_{D,D'}$ which is the optimal δ *with respect to a single, fixed pair of (neighboring) databases* D, D'. Looking ahead in the next section, we will first tackle the easier problem of estimating $\delta_{D,D'}$ (Section 5.1), before tackling the harder problem of estimating δ itself (Section 5.2).

*Definition 4.1 (**Optimal** δ).* Let $\mathcal{M}$ be a mechanism, D, D' be databases, and $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. We say the privacy parameter $\delta_{D,D'}$ is optimal (minimal) with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$ if

$$\delta_{D,D'} = \max(\max_{\mathcal{S} \subseteq O} \Pr[\mathcal{M}(D) \in \mathcal{S}] - e^{\varepsilon} \Pr[\mathcal{M}(D') \in \mathcal{S}], 0).$$

We say the privacy parameter δ is optimal (minimal) with respect to the tuple $(\mathcal{M}, \varepsilon)$ if

$$\delta = \max_{D \simeq D'} \left\{ \max(\delta_{D,D'}, \delta_{D',D}) \right\}.$$

Then, we define a (perfect) DP estimator, which, given a mechanism $\mathcal{M}$ and one of the privacy parameters ε, outputs the optimal δ such that $\mathcal{M}$ is (ε, δ)-DP.

*Definition 4.2 (**Perfect DP Estimator**).* Let $\mathcal{C} = \mathcal{X} \mapsto \mathcal{O}$ be the set of poly$(\log |\mathcal{X}|)$-time mechanisms, $\mathcal{M} \in \mathcal{C}$ be a mechanism from the set $\mathcal{C}$, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. An algorithm is a Perfect DP Estimator for $\mathcal{C}$, if for every $(\mathcal{M}, \varepsilon)$, with black-box access to $\mathcal{M}$, the algorithm outputs the optimal δ with respect to the tuple $(\mathcal{M}, \varepsilon)$.

Unfortunately, a perfect DP estimator does not exist. In fact, we can show something even stronger—even an approximate version of a DP estimator (Def. 4.4) still does not exist (Theorem 4.5). Intuitively, this is because a general estimator would need to test the DP property for all pairs of databases—an impossible task for a polynomial-time algorithm if the number of databases in the mechanism's domain is super-polynomial. The proof of the theorem follows the above intuition and can be found in Appendix A.

*Definition 4.3 (α-**tight bound**).* Let $\mathcal{M}$ be a mechanism, D, D' be databases, and $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. We say $\delta'_{D,D'}$ is a α-tight bound with respect to $(\mathcal{M}, D, D', \varepsilon)$ if

$$|\delta'_{D,D'} - \delta_{D,D'}| \leqslant \alpha,$$

where $\delta_{D,D'}$ is optimal with respect to $(\mathcal{M}, D, D', \varepsilon)$.

Similarly, we say δ' is a α-tight bound with respect to $(\mathcal{M}, \varepsilon)$ if

$$|\delta' - \delta| \leqslant \alpha,$$

where δ is optimal with respect to $(\mathcal{M}, \varepsilon)$.

*Definition 4.4 (**Approximate DP Estimator**).* Let $\mathcal{C} = \mathcal{X} \mapsto \mathcal{O}$ be the set of poly$(\log |\mathcal{X}|)$-time mechanisms, $\mathcal{M} \in \mathcal{C}$ be a mechanism from the set $\mathcal{C}$, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. An algorithm is a (α, β)-Approximate DP Estimator for $\mathcal{C}$, if for every $(\mathcal{M}, \varepsilon)$, with black-box access to $\mathcal{M}$, with probability at least $1 - \beta$, it provides α-tight bound with respect to the tuple $(\mathcal{M}, \varepsilon)$, where $\alpha, \beta \in [0, 1)$.

Theorem 4.5. *Let $\alpha \in [0, \frac{1}{2})$ and $\beta \geqslant \frac{1}{2} + \nu(n)$, where ν is a non-negligible function. Let $\mathcal{C} = \{0,1\}^n \mapsto \mathcal{O}$ be the set of* poly(n)*-time mechanisms. There doesn't exist a* poly(n)*-time (α, β)-Approximate DP Estimator for $\mathcal{C}$.*

One can verity that the above impossibility also applies to common relaxations of DP from the literature, such as Renyi DP [9] and (Zero-)Concentrated DP [10, 11]. Intuitively, the reason is the following: if to test a mechanism's property (in the worse case) we need to test the property for all pairs of the mechanism's input, and the number of pairs is unbounded, then we cannot have an efficient algorithm for this task. This intuition, which is at the core of the proof of Theorem 4.5, applies also to the above variants, and points to the idea that in order to circumvent our impossibility, it seems necessary to bound the size of the mechanism's input space, which motivates the relative-DP relaxation detailed in the following.

4.1 Relative Differential Privacy

In view of the impossibility stated in Theorem 4.5, we ask: "Is there a meaningful/useful relaxation to differential privacy that allows us to circumvent this impossibility?" Using a similar argument as Thm. 4.5, we can also show that the impossibility also applies to well-known relaxations of DP such as Renyi DP [9] and (Zero-)Concentrated DP [10, 11]. To answer the above question in affirmative, we introduce *relative differential privacy*, which we believe is a minimal (in terms of intuitive distance from DP) and useful definition. Relative DP considers the privacy of a mechanism *relative to a set of databases*. As discussed in our introduction, this models the case where the mechanism will only be applied to a limited number of databases, such as the database of census results in 2020 Census in the United States [34]. Informally, a mechanism is $(\varepsilon, \delta, \mathcal{T})$-relative DP if on domain restricted to $\mathcal{T}$, the mechanism is (ε, δ)-DP.

Recall, we defined 'neighboring' (Def. 3.2) as 'remove one row' rather than 'remove-or-add one row', so that the number of neighbors of a database does not depend on the domain of each database row. This modification did not change the original DP definition, but allows our Thm 5.10 to circumvent impossibility Thm 4.5 for superpolynomial-size domains in our *relative DP* definition.

*Definition 4.6 ($\varepsilon, \delta, \mathcal{T}$-**relative Differential Privacy**).* A mechanism $\mathcal{M} \coloneqq \mathcal{X} \mapsto \mathcal{O}$ is $(\varepsilon, \delta, \mathcal{T})$-relative differentially private if for all subset $\mathcal{S} \subseteq \mathcal{O}$ and all neighboring databases $D \simeq D' : D \in \mathcal{T}$:

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

and

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

where the probability space is over the coin flips of mechanism $\mathcal{M}$.

To further motivate the definition of relative DP, we also show it satisfies several useful properties (such as composition (Prop. 4.9, and 4.10) and post-processing (Prop. 4.11)), that are comparable to those of classical DP. The proofs of the following propositions can be found in Appendix B.

It is clear to see that relative DP and DP are the same, if $\mathcal{T}$ is the same as the domain of the mechanism. Moreover, a mechanism that is private for $\mathcal{T}_1$ and $\mathcal{T}_2$ is also private for $\mathcal{T}_1 \cup \mathcal{T}_2$ ($\mathcal{T}$ scalable).

Proposition 4.7. *If the mechanism $\mathcal{M}$ is $(\varepsilon, \delta, \mathcal{T})$-relative differentially private and $\mathcal{T} = \mathcal{X}$, then the mechanism $\mathcal{M}$ is (ε, δ)-differentially private.*

One might be worried that by providing such a relative version of DP, we might be creating a privacy notion that melts down once new databases are added to the mix. The following proposition shows that this is not the case for relative DP, as long as the mechanism behaves well on the new database. Note that this requirement also exists in DP, where parallel composition also takes the max of the privacy of all composed mechanisms.

Proposition 4.8. *[$\mathcal{T}$ **Scalable**] If the mechanism $\mathcal{M}$ is $(\varepsilon_1, \delta_1, \mathcal{T}_1)$-relative differentially private, $\cdots$, and $(\varepsilon_k, \delta_k, \mathcal{T}_k)$-relative differentially private, then the mechanism is also $\left(\max_{i\in[k]} \varepsilon_i, \max_{i\in[k]} \delta_i, \bigcup_{i\in[k]} \mathcal{T}_i\right)$- relative DP.*

Relative DP also enjoys the same convenient guarantees as DP: parallel composition, sequential composition, as well as post-processing.

Proposition 4.9. *[**Parallel Composition**] Let $\mathcal{T}_1 \times \mathcal{T}_2$ be the concatenation of set $\mathcal{T}_1$ and $\mathcal{T}_2$, that is, $\mathcal{T}_1 \times \mathcal{T}_2 = \{(D_1, D_2) : D_1 \in \mathcal{T}_1 \wedge D_2 \in \mathcal{T}_2\}$. If $\mathcal{M}_1, \cdots, \mathcal{M}_k$ are k mechanisms, where $\mathcal{M}_i$ satisfies $(\varepsilon_i, \delta_i, \mathcal{T}_i)$-relative differential privacy, then the mechanism $\mathcal{M}$ taking database $(D_1, \cdots, D_k) \in \mathcal{T}_1 \times \cdots \times \mathcal{T}_k$ as inputs and outputting $(\mathcal{M}_1(D_1), \cdots, \mathcal{M}_k(D_k))$ is $\left(\max_{i\in[k]} \varepsilon_i, \max_{i\in[k]} \delta_i, \mathcal{T}_1 \times \cdots \times \mathcal{T}_k\right)$-relative DP.*

Proposition 4.10. *[**Sequential Composition**] If $\mathcal{M}_1, \cdots, \mathcal{M}_k$ are k mechanisms, where $\mathcal{M}_i$ satisfies $(\varepsilon_i, \delta_i, \mathcal{T})$- relative differentially privacy, then the mechanism $\mathcal{M} := (\mathcal{M}_1, \cdots, \mathcal{M}_k)$ is $\left(\sum_{i\in[k]} \varepsilon_i, \sum_{i\in[k]} \delta_i, \mathcal{T}\right)$- relative DP.*

Proposition 4.11. *[**Post-processing**] If $\mathcal{M}_1$ is a mechanism that satisfies $(\varepsilon, \delta, \mathcal{T})$-relative differentially privacy, then for any (randomized) algorithm f, the mechanism $\mathcal{M} := f(\mathcal{M}_1)$ is $(\varepsilon, \delta, \mathcal{T})$-relative differentially private.*

5 (RELATIVE) DP ESTIMATOR

In this section we define and analyse our (relative) privacy estimator. As discussed in the introduction, we start (in Section 5.1) with the simple case of $|\mathcal{T}| = 1$ and in particular with one fixed pair of databases. Although this is clearly not particularly relevant for a general privacy definition, it still offers an interesting ball field for introducing our main ideas, and allows us a smooth transition to our general estimator which is described and analyzed in Section 5.2.

5.1 Estimating δ for a pair of databases

As the first step in defining our privacy estimator, we narrow the definition of a privacy estimator to define a privacy estimator for a single pair of neighboring databases. We construct a class of concrete privacy estimator algorithms $\mathcal{A}^{\mathsf{B}}_C$ by relating the privacy parameter δ to the *risk* (or error) of a classification algorithm B (Theorem 5.4). Inheriting tight bounds on risk from the classification algorithm's convergence theorem, we show in Theorem 5.7 (using the kNN classification algorithm as example), that our privacy estimator algorithm also enjoys tight accuracy bounds.

Our results in this section show that, despite the impossibility of general DP estimator and the lack of tight bounds in previous work, it is indeed possible to construct relative DP estimators with tight accuracy bounds. In the next section, we will extend algorithm $\mathcal{A}^{\mathsf{B}}_C$ of this section to construct a privacy estimator for any $(\varepsilon, \delta, \mathcal{T})$-relative DP mechanism.

5.1.1 Privacy Estimator for a Pair of Databases. First, we define a perfect δ estimator for a pair of databases. Informally, this estimator must always output the optimal δ (see Def. 4.1).

*Definition 5.1 (**Perfect δ-Estimator for a Pair of Databases**).* Let $C = \mathcal{X} \mapsto O$ be the set of poly$(\log|\mathcal{X}|)$-time mechanisms. $\mathcal{M} \in C$ be a mechanism from the set C. D, D' be databases, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. An algorithm is a Perfect δ-Estimator for a Pair of Databases for C if for every $(\mathcal{M}, D, D', \varepsilon)$ with black-box access to $\mathcal{M}$, the algorithm outputs the optimal $\delta_{D,D'}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$.

However, a perfect estimator for a pair of databases does not exist—by our Theorem 5.4 below, a perfect estimator would imply the existence of an optimal classifier achievable with limited training samples. Thus, we define below an approximate estimator Def. 5.2, with similar approximation parameters α and β as for the approximate DP privacy estimator Def. 4.4.

*Definition 5.2 (**Approximate δ-Estimator for a Pair of Databases**).* Let $C = \mathcal{X} \mapsto O$ be the set of poly$(\log|\mathcal{X}|)$-time mechanisms, $\mathcal{M} \in C$ be a mechanism from the set C, D, D' be databases, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. An algorithm is a (α, β)-Approximate δ-Estimator for a Pair of Databases for C if for every $(\mathcal{M}, D, D', \varepsilon)$, with black-box access to $\mathcal{M}$, with probability at least $1 - \beta$, it provides α-tight bound with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$, where $\alpha, \beta \in [0, 1)$.

5.1.2 Relating Privacy Parameter δ to Risk of the Bayes Classifier. Now we have defined an approximate privacy estimator with respect to a pair of databases (Def. 5.2), we present our construction of such an estimator. The basis of our estimator is a connection between the definition of DP and the *risk* of a Bayes Classifier, described in Theorem 5.4 below.

For a mechanism $\mathcal{M}$, a database D, and privacy parameter ε, let $\left[\mathcal{M}(D)\right]_\varepsilon$ denote the random variable obtained by tossing a biased coin c where $\Pr[c = 1] = e^{-\varepsilon}$, and receiving value $\mathcal{M}(D)$ if $c = 1$ or receiving value $\perp$ (a null value not in the range of $\mathcal{M}$) otherwise.

Definition 5.3 (The distribution $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$). Let $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ denote the distribution of a random variable, which is obtained by tossing a fair coin b, and receiving tuple $(\mathcal{M}(D'), 1)$ if $b = 1$ or receiving value $(\left[\mathcal{M}(D)\right]_\varepsilon, 0)$ otherwise.

The proof of the theorem below (App. C) is based on the fact that δ in (ε, δ)-(relative) DP can be re-written in terms of a *statistical distance*[8] between two random variables. The difference between the DP definition and statistical distance is that in DP, one of the probabilities is scaled by e^{ε}. This means we can re-write $\delta_{D,D'}$ in terms of the statistical distance between two r.v.'s $\mathcal{M}(D')$ and $\left[\mathcal{M}(D)\right]_{\varepsilon}$ (which, intuitively, 'scales' the distribution of $\mathcal{M}(D)$ by $1/e^{\varepsilon}$). Then, the theorem follows from the connection between statistical distance and the accuracy (or risk) of the optimal (or Bayes) classifier.

Theorem 5.4 (Mechanism Privacy as Bayes Classifier Risk). *Let $\mathcal{M}$ be a mechanism, D, D' be databases, and $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. Let $h^*_{D,D'}$ be the Bayes classifier for $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ (Def. 5.3, abbreviated as $\mathcal{P}$ below). The optimal delta $\delta_{D,D'}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$ satisfies the following equality*

$$\delta_{D,D'} = \max\left(1 - 2e^{\varepsilon} R(h^*_{D,D'}), 0\right),$$

Corollary 5.5. *Let the mechanism $\mathcal{M}$, privacy parameter ε, distribution $\left[\mathcal{M}(D)\right]_{\varepsilon}$, and the Bayes classifier $h^*_{D,D'}$ defined the same as that in Theorem 5.4. Let $D \simeq D'$ be a neighboring databases pair. The optimal δ with respect to the tuple $(\mathcal{M}, \varepsilon)$ satisfies the equality*

$$\delta = \max_{D \simeq D'} \left\{\max(1 - 2e^{\varepsilon} R(h^*_{D,D'}), 1 - 2e^{\varepsilon} R(h^*_{D',D}), 0)\right\}$$

5.1.3 Privacy Estimator for Neighboring Databases with Tight Accuracy Bounds. In this section, we take advantage of the connection between DP and the risk of the Bayes classifier (Theorem 5.4), to construct an approximate DP estimator for a single pair of databases (see Def. 5.2). Our algorithm $\mathcal{A}^{\mathsf{B}}_{C}$, Fig. 1, is parameterized by any classifier B, and generates a privacy estimate via the computed risk of this classifier.

Lemma 5.6 (Proof in Appendix D). *Let $C = \mathcal{X} \mapsto O$ be the set of $\mathrm{poly}(\log|\mathcal{X}|)$-time mechanisms, $\mathcal{M} \in C$ be a mechanism from the set C, D, D' be databases, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. Let $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ be as in Def. 5.3, abbreviated as $\mathcal{P}$. Let $h^*_{D,D'}$ be the Bayes classifier for $\mathcal{P}$. Let h^{B}_n be a classifier for $\mathcal{P}$ produced by binary classification algorithm B with n samples. Let $g(\mathcal{X}, n, \beta)$ be a function of input space $\mathcal{X}$, sample size n and $\beta \in (0,1)$.*

If for every $(\mathcal{M}, D, D', \varepsilon)$, where $\mathcal{M} \in C$, with probability at least $1-\beta$, we have $|R(h^{\mathsf{B}}_n) - R(h^*_{D,D'})| = O\left(g(\mathcal{X}, n, \beta)\right)$, then the algorithm $\mathcal{A}^{\mathsf{B}}_{C}$ with n samples, shown in Figure 1 , is a (α, β)-Approximate δ-Estimator for a Pair of Databases for C, for any $\alpha = O\left(g(\mathcal{X}, n/2, \beta/2) + \sqrt{\ln(1/\beta)/n}\right)$, $\beta \in (0,1)$, $c \in \mathsf{R}$.

We state the theorem for the case where our classifier is kNN. On a more technical note, we remark that Thm. 5.7 does not contradict the impossibility results from Antos *et al.* [14]. In fact, our theorem uses as blackbox Thm 11.1 in Devroye et al. [12] which only requires a density. Similar to [12], our theorem statement is asymptotic.

Theorem 5.7 (Proof in Appendix E). *Consider the set of mechanisms $C = \mathcal{X} \mapsto \mathsf{R}^d$ whose output distributions have a density. kNN is the kNN classification algorithm with n samples where $k = \sqrt{n}$. The algorithm $\mathcal{A}^{\mathsf{kNN}}_{C}$, shown in Figure 1, is a (α, β)-Approximate δ-Estimator for a Pair of Databases for C, for any $\alpha = O\left(c_d\sqrt{\ln(1/\beta)/n}\right)$, $\beta \in (0,1)$, where $c_d \leqslant (1 + 2/\sqrt{2-\sqrt{3}})^d - 1 \leqslant 4.86371^d$ (Lemma 5.5, [12]).*

[8] Statistical distance between two r.v. X, Y is defined as $\Delta(X,Y) = \max_S |\Pr(X \in S) - \Pr(Y \in S)|$.

5.2 Estimating Approximate Relative DP

In this section, we extend our algorithm from our previous section, to construct a privacy estimator for Relative Differential Privacy (Def. 4.6). We begin with a formal definition for a relative DP estimator with tight bounds (Def. 5.9). Then, we present our privacy estimator which builds upon algorithm $\mathcal{A}^{\mathsf{B}}_{C}$ from Section 5.1.3. Given any classification algorithm B, our privacy estimator $\mathcal{A}^{B}_{C,t}$ outputs the privacy parameter for any mechanism in class C and set of databases of size t. Using the kNN classifier as example, we show in Thm. 5.10 that our privacy estimator indeed satisfies tight accuracy bounds.

5.2.1 Our Approximate Relative DP Estimator. Before describing our DP estimator, we first define the guarantees such a (α, β)-approximate relative DP estimator should satisfy. Intuitively, these are the same as for an approximate DP estimator, except we restrict the domain of our mechanism to the set $\mathcal{T}$, relative to which we define privacy.

Definition 5.8. Let $\mathcal{M}$ be a mechanism, $\mathcal{T} \subseteq \mathcal{X}$ be a set of databases, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter, $D \simeq D'$ be a pair of neighboring databases. We say the privacy parameter $\delta_{\mathcal{T}}$ is optimal with respect to $(\mathcal{M}, \mathcal{T}, \varepsilon)$, if

$$\delta_{\mathcal{T}} = \max_{\substack{D \in \mathcal{T}: \\ D \simeq D'}} \left\{\max(\delta_{D,D'}, \delta_{D',D})\right\},$$

where $\delta_{D,D'}$ is optimal with respect to $(\mathcal{M}, D, D', \varepsilon)$. We say $\delta'_{\mathcal{T}}$ is an α-tight bound with respect to $(\mathcal{M}, \mathcal{T}, \varepsilon)$, if $|\delta'_{\mathcal{T}} - \delta_{\mathcal{T}}| \leqslant \alpha$.

*Definition 5.9 (**Approximate Relative DP Estimator**).* Let $C = \mathcal{X} \mapsto O$ be the set of $\mathrm{poly}(\log|\mathcal{X}|)$-time mechanisms, $\mathcal{M} \in C$ be a mechanism from the set C, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. Let $\mathcal{T} \subseteq \mathcal{X}$ be any set of databases. An algorithm is a (α, β)-*Approximate Relative DP Estimator* for C if for every $(\mathcal{M}, \mathcal{T}, \varepsilon)$ with black-box access to $\mathcal{M}$ with probability at least $1-\beta$, it provides α-tight bound with respect to the tuple $(\mathcal{M}, \mathcal{T}, \varepsilon)$ for any $\alpha, \beta \in [0,1)$.

We are now ready to formally define and analyze our Algorithm, denoted as $\mathcal{A}^{B}_{C,t}$ (see Fig. 2 for a detailed description). $\mathcal{A}^{B}_{C,t}$ uses our estimator for pairs of neighboring databases (see Fig. 1) and runs it for all neighbors of set $\mathcal{T}$. Intuitively, by union bound, our accuracy degrades multiplicatively with the total number of neighbors of databases in $\mathcal{T}$. This leads to our main Theorem 5.10 that shows the accuracy of our privacy estimator based on the kNN classifier.

Theorem 5.10 ((α, β)-Approximate Relative DP Estimator, using kNN, Proof in Appendix F). *Consider the set of mechanisms $C = \mathcal{U}^m \mapsto \mathsf{R}^d$ whose output distribution has a density. Let $\mathcal{T} \subseteq \mathcal{X}$ be any set of databases in relative DP, $|\mathcal{T}| \leqslant t$. Let the algorithm B be $\mathcal{A}^{\mathsf{kNN}}_{C}$ with n samples, shown in Figure 1. The algorithm $\mathcal{A}^{B}_{C,t}$, shown in Figure 2, is a (α, β)-Approximate Relative DP Estimator for C, where $\alpha = O\left(c_d\sqrt{\ln(2tm/\beta)/n}\right)$, $\beta \in (0,1)$.*

6 DISTRIBUTIONAL DIFFERENTIAL PRIVACY

As an extension of our results, we present the *first* privacy estimator for $(\varepsilon, \delta, \Delta)$-*distributional differential privacy* (Def. 3.5), given Δ

Input: A binary classification algorithm B with n samples. A mechanism $\mathcal{M} \in C$, a pair of databases $D, D' \in \mathcal{X}$, privacy parameter $\varepsilon \in \mathsf{R}_{\geqslant 0}$.
Output: $\delta'_{D,D'}$, the estimate of the optimal delta $\delta_{D,D'}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$.
Recall $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ (Def. 5.3, abbreviated below as $\mathcal{P}$) denotes the distribution of a random variable, which is obtained by tossing a fair coin b, and receiving tuple $(\mathcal{M}(D'), 1)$ if $b = 1$ or receiving value $(\left[\mathcal{M}(D)\right]_\varepsilon, 0)$ [a] otherwise.

(1) Initialize $n_1 \leftarrow n/2$, $n_2 \leftarrow n/2$, and $r \leftarrow 0$.
(2) Sample n_1 training points $(o_1, b_1), \cdots, (o_{n_1}, b_{n_1})$ according to joint distribution $\mathcal{P}$.
(3) Taking the n_1 training points as inputs, classification algorithm B outputs a classifier $h^{\mathsf{B}}_{n_1}$.
(4) Repeat the process n_2 times: ▷ Estimate risk function of classifier $h^{\mathsf{B}}_{n_1}$ with n_2 testing samples.
    (a) Sample a testing point (o, b) according to joint distribution $\mathcal{P}$.
    (b) Predict the sample o's label using the trained classifier: $b' = h^{\mathsf{B}}_{n_1}(o)$. If $b' \neq b$, $r \leftarrow r + 1/n_2$.
(5) Output $\delta'_{D,D'} \leftarrow \max(1 - 2e^{\varepsilon} r, 0)$.

[a] Recall $\left[\mathcal{M}(D)\right]_\varepsilon$ is a distribution for tossing a coin c where $\Pr[c = 1] = e^{-\varepsilon}$, outputting $\mathcal{M}(D)$ if $c = 1$ or $\perp$ (a null value) otherwise.

Figure 1: $\mathcal{A}^{\mathsf{B}}_{C}$, an algorithm for estimating the optimal delta with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$

Input: An algorithm B with n samples, which estimates the optimal $\delta_{\mathcal{T}}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$ for mechanism family C. A mechanism $\mathcal{M} \in C$, a set of databases $\mathcal{T}$, privacy parameter $\varepsilon \in \mathsf{R}_{\geqslant 0}$.
Output: $\delta'_{\mathcal{T}}$, the estimate of the optimal delta $\delta_{\mathcal{T}}$ with respect to the tuple $(\mathcal{M}, \mathcal{T}, \varepsilon)$.

(1) For each neighboring databases $D \simeq D'$ where $D \in \mathcal{T}$, use algorithm B with n samples compute the estimate of $\delta_{D,D'}$ and the estimate of $\delta_{D',D}$. Denote the maximum among these estimates as $\delta'_{\mathcal{T}}$.
(2) Output $\delta'_{\mathcal{T}}$.

Figure 2: $\mathcal{A}^{B}_{C,t}$, an algorithm for estimating the optimal delta with respect to the tuple $(\mathcal{M}, \mathcal{T}, \varepsilon)$

contains database distributions where each entry is independently distributed. Of importance, by considering databases as random variables that model a level of adversarial uncertainty about the data, DDP—unlike DP—can formally measure the privacy of even deterministic mechanisms. This means, for the first time, we have shown a method to heuristically estimate the privacy of deterministic mechanisms (under independently distributed data).

First, we observe that DDP under the independence assumption (Def. 3.5) is very similar to DP. This allows us to define an approximate privacy estimator in a similar manner.

Definition 6.1. Let $\mathcal{M}$ be a mechanism, $D \simeq D'$ be a pair of neighboring databases, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter, and Δ be a set of distributions on size-m databases where each row is independently distributed. We say the privacy parameter δ_{DDP} is optimal with respect to the tuple $(\mathcal{M}, \Delta, \varepsilon)$ if

$$\delta_{\mathsf{DDP}} = \max\Big(\max_{\pi \in \Delta, i \in [m], x, x' \in \mathcal{U}, S \subseteq O} \Pr_{D \sim \pi}[\mathcal{M}(D) \in S | D_i = x] - e^{\varepsilon} \Pr_{D \sim \pi}[\mathcal{M}(D) \in S | D_i = x'],\ 0 \Big).$$

We say δ'_{DDP} is a α-tight bound with respect to $(\mathcal{M}, \Delta, \varepsilon)$, if

$$|\delta'_{\mathsf{DDP}} - \delta_{\mathsf{DDP}}| \leqslant \alpha.$$

*Definition 6.2 (**Approximate DDP Estimator**).* Let $C = \mathcal{X} \mapsto O$ be the set of poly$(\log |\mathcal{X}|)$-time mechanisms, $\mathcal{M} \in C$ be a mechanism from the set C, $\varepsilon \in \mathsf{R}_{\geqslant 0}$ be a privacy parameter. Let Δ be any set of distributions on size m databases, such that $|\Delta| \leqslant t$ for some $t \in \mathcal{N}^+$. An algorithm is a (α, β)-Approximate DDP Estimator for C if for every $(\mathcal{M}, \Delta, \varepsilon)$, with black-box access to $\mathcal{M}$, with probability at least $1 - \beta$, it provides α-tight bound with respect to the tuple $(\mathcal{M}, \Delta, \varepsilon)$, where $\alpha, \beta \in [0, 1)$, and $|\Delta| \leqslant t$.

Our DDP estimator $\mathcal{A}^{B}_{C,\Delta}$, described formally in Fig. 3, is essentially the same as our relative DP estimator, except it is even simpler—here, we only need to run our estimator on the distributions in Δ, rather than enumerating all databases in $\mathcal{T}$. The accuracy of $\mathcal{A}^{B}_{C,\Delta}$ is thus a corollary of Theorem 5.10.

Corollary 6.3. *Consider the set of mechanisms $C = \mathcal{U}^m \mapsto \mathsf{R}^d$ whose output distribution has a density. Let the algorithm B be $\mathcal{A}^{\mathsf{kNN}}_{C}$ with n samples, shown in Fig. 2. The algorithm $\mathcal{A}^{B}_{C,\Delta}$, shown in Figure 3, is a (α, β)-Approximate DDP Estimator for C, where $\alpha = O(c_d \sqrt{\ln(mt|\mathcal{U}|^2/\beta)/n})$, $\beta \in (0, 1)$.* [9]

7 VALIDATION AND BENCHMARKING

We next demonstrate the applicability of our theoretical construction and the accuracy of the theory presented above. To do so, we have devised a proof-of-concept implementation of our estimator which we use in two different modes: First we focus on the two most common DP mechanisms, the *Laplacian mechanism* and the *Gaussian mechanism*, for which we have well understood theory yielding analytical bounds that we can compare our estimator's output against. Informally, these two mechanisms achieve differential privacy by adding noise drawn from Laplace (resp. Gaussian) distribution to query results. In particular, Gaussian mechanism is one of the most important building blocks to achieve (ε, δ)-DP, and as far as we know, our work is the first to test our heuristic estimator on this mechanism.

Second, we benchmark our implementation against Sparse Vector Technique (SVT), a fundamental differential privacy mechanism which takes a sequence of queries $\mathcal{Q}$ and a sequence of threshold $\mathcal{T}$ as input, and outputs a Boolean vector indicating whether each query over the database is above or below the corresponding threshold in $\mathcal{T}$. We note that SVT is a more complex mechanism for which no exact analytical privacy bound is known. Nonetheless, it serves

[9] Recall that $\mathcal{U}$ is the space of values each entry in the database can take (see Def. 3.1).

Input: A binary classification algorithm B with n samples, mechanism $\mathcal{M} \in C$, privacy parameter $\varepsilon \in \mathsf{R}_{\geqslant 0}$, and set of distributions Δ.
Output: δ'_{DDP}, the estimate of the optimal delta δ_{DDP} with respect to the tuple $(\mathcal{M}, \Delta, \varepsilon)$.

Let $X_{x,i,\pi}$ denote the random variable outputting by the following experiment: sample a database D according to distribution π. Set the i-th row of D to records x. Return $\mathcal{M}(D)$.

Let $\left[X_{x,i,\pi}\right]_{\varepsilon}$ denote the random variable obtained by tossing a biased coin c where $\Pr[c = 1] = e^{-\varepsilon}$, and receiving value $X_{x,i,\pi}$ if $c = 1$ or receiving value $\perp$ (a null value not in the range of $\mathcal{M}$) otherwise.

Let $\mathcal{P}$ denote the distribution of a random variable, which is obtained by tossing a fair coin b, and receiving tuple $(X_{x',i,\pi}, 1)$ if $b = 1$ or receiving value $(\left[X_{x,i,\pi}\right]_{\varepsilon}, 0)$ otherwise.

(1) Initialize $n_1 \leftarrow n/2$, $n_2 \leftarrow n/2$, and $\delta'_{\mathsf{DDP}} \leftarrow 0$.
(2) For all $\pi \in \Delta, i \in [m], x, x' \in \mathcal{U}$
  (a) Initialize $r \leftarrow 0$.
  (b) Sample n_1 training points $(o_1, b_1), \cdots, (o_{n_1}, b_{n_1})$ according to joint distribution $\mathcal{P}$.
  (c) Taking the n_1 training points as inputs, classification algorithm B outputs a classifier $h^{\mathsf{B}}_{n_1}$.
  (d) Repeat the process n_2 times: ▷ Estimate risk function of classifier $h^{\mathsf{B}}_{n_1}$ with n_2 testing samples.
    (i) Sample a testing point (o, b) according to joint distribution $\mathcal{P}$.
    (ii) Predict the sample o's label using the trained classifier: $b' = h^{\mathsf{B}}_{n_1}(o)$. If $b' \neq b$, $r \leftarrow r + 1/n_2$.
  (e) Update $\delta'_{\mathsf{DDP}} \leftarrow max\left(\delta'_{\mathsf{DDP}}, 1 - 2e^{\epsilon} r\right)$.
(3) Output $\delta'_{D,D'}$.

Figure 3: $\mathcal{A}^{B}_{C,\Delta}$, an algorithm for estimating the optimal delta δ_{DDP} with respect to the tuple $(\mathcal{M}, \Delta, \varepsilon)$

as a perfect benchmark as (1) we can still compare our results to the state of the art implementation [8], and (2) the literature offers alternative implementations of SVT, some of which are known to be buggy [17] which can be used to demonstrate the ability of our estimator to compare the quality of different mechanisms.

We complete the section with two further applications of our theory, namely comparing different implementations of DP mechanisms and verifying an implementation, demonstrating how our system can be used to solve problems in DP that have attracted a lot of attention in recent security literature.

7.1 Benchmarking and Validating our Theory

Our first two sets of experiments estimate the privacy parameters of the common Laplacian and Gaussian mechanisms, denoted as $\mathcal{M}_{L,\varepsilon}$ and $\mathcal{M}_{G,\varepsilon,\delta}$ respectively (We recall these mechanisms in Definitions G.1 and G.2 in appendix G.)

Knowing just a single pair of privacy parameters (ε, δ) for a mechanism may be insufficient to understand its privacy guarantees. It does not answer, for example, the question "What happens to δ (resp. ε) if I claim a smaller ε (resp. δ) for the same mechanism?". This question can be answered by understanding how the claimed ε (the privacy achieved) for this mechanism affects its associated δ (probability of privacy failure). In Figures 4a and 4b, we use our privacy estimator to plot, for $\mathcal{M}_{L,\varepsilon}$ and $\mathcal{M}_{G,\varepsilon,\delta}$, the privacy parameter ε against its corresponding optimal δ (Def. 4.1). The figures show the accuracy of our estimate of δ to the analytically computed optimal δ (see Lemma G.3 and Lemma G.4), demonstrating that our estimator not only enjoys tight theoretical accuracy bounds, it also achieves even better experimental accuracy.

Our second set of experiments on SVT demonstrates that the DP spectrum computed by our estimator (Fig. 4c) is comparable with the state of the art ([8], Fig. 1e, e.g., around $\delta = 0.055$ for $\varepsilon = 0$ for SVT). Note that, whereas [8] is specialized for mechanisms with smaller output space, our estimator works with large output spaces as well; to our knowledge ours the first black-box (ε, δ) privacy estimator with this property.

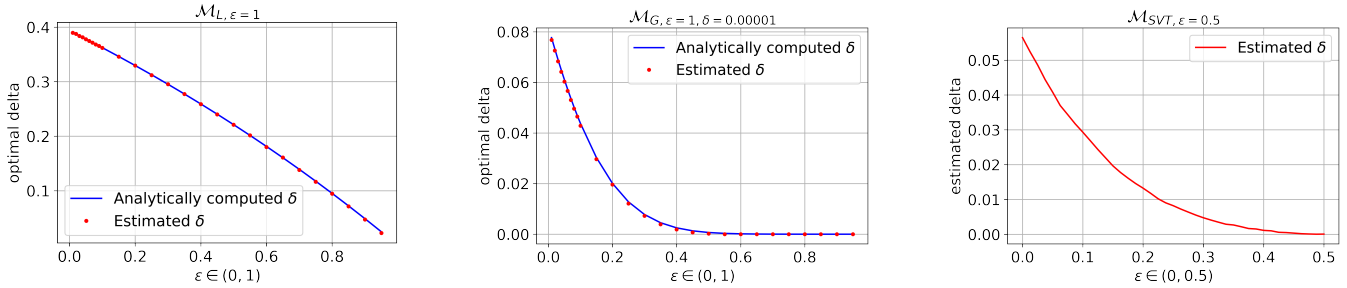
Figure 5 plots the number of samples used in our kNN-based privacy estimator, against the guaranteed α parameter (recall from Def. 4.3, this describes the accuracy of our estimator output). The tested mechanism is the noised bit query Laplace mechanism $\mathcal{M}_{L,\varepsilon}$ (with sensitivity 1). We use the empirical bootstrapping method, run the estimator 30 times and set the confidence interval as 0.9. From the figure, we see that the empirical α is 3 orders of magnitude tighter than the theoretical α. When the number of sample points is 2^{26}, (about 10 minutes running time on a Dell compute node with two 64-core AMD Epyc 7662 "Rome" processors and 256 GB memory) the estimated δ is within an additive error less than 0.0001, which is also shown in Figure 4a and Figure 4b.

The above demonstrates that our implementation tightly matches the theory developed in our framework (and at a level impressive for machine learning applications). On the one hand, this establishes the usefulness of our framework and implementation as a very accurate privacy estimator; (2) on the other hand, our experiments on SVT demonstrates that our estimator, even in this proof-of-concept implemenation, can be applied to more complex mechanisms, serving as evidence of its potential practical usage.

Remark. We talk about additive error in two ways: the one theoretically derived from Thm. 5.10 and the one computed via experiment. For the theoretically computed one, we fix failure probability to 0.01 and then compute the additive error according to Thm. 5.10. For the experimental one, we use the empirical bootstrap method to compute a confidence interval (CI) for a 0.9 confidence level. Hence, 0.9 is the probability we are within the CI and the length of the CI corresponds to the additive error.

7.2 Further Applications

In the remainder of this section, we showcase two additional useful applications of our privacy estimation framework: (1) To compare

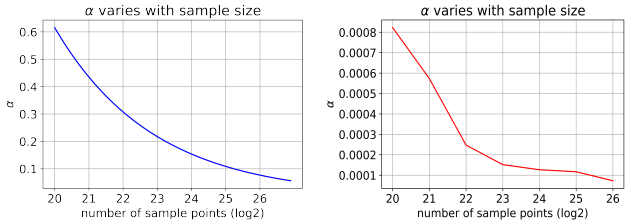


(a) Analytical computed optimal δ compared with estimated δ for $\mathcal{M}_{L,\varepsilon}$

(b) Analytical computed optimal δ compared with estimated δ for $\mathcal{M}_{G,\varepsilon,\delta}$

(c) Estimated (ε, δ)-spectrum of $\mathcal{M}_{\text{SVT},\varepsilon}$ parameterized with $\varepsilon = 0.5, \delta = 0$.

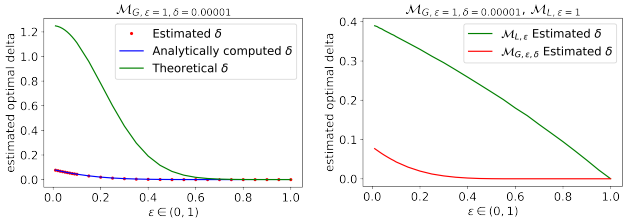
Figure 4: Accuracy check for our DP estimator implementation



(a) Theoretical α

(b) Empirical α

Figure 5: Left: Theoretical accuracy α of estimated δ vs. number of samples (Theorem 5.7). Right: empirical accuracy for $\mathcal{M}_{L,\varepsilon}$.



(a) Loose theoretical (top curve) vs. actual (analytically computed and estimated) (ε, δ)-privacy spectrum of $\mathcal{M}_{G,\varepsilon,\delta}$

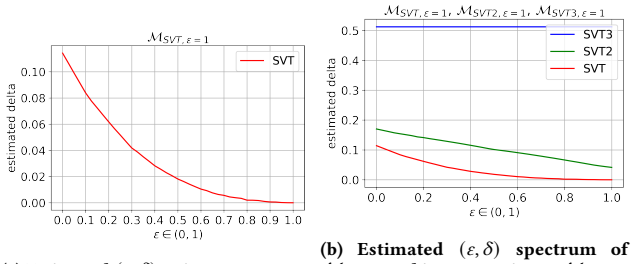
(b) Estimated (ε, δ)-privacy spectrum of $\mathcal{M}_{G,\varepsilon,\delta}$ and $\mathcal{M}_{L,\varepsilon}$. We see $\mathcal{M}_{G,\varepsilon,\delta}$ (bottom curve) achieves better (smaller) δ.

Figure 6: Application 1: Comparing mechanism privacy?

what we term the *(differential) privacy spectrum* (i.e., the tradeoff between ε and δ) of two mechanisms, and (2) to verify the implementation of a given mechanism. We note in passing, that as discussed above, a major application of our method is for estimating the privacy of heuristic approaches to privatizing machine learning algorithms. We view this as a very promising research direction, albeit beyond the scope of this work which aims at introducing, analyzing, and validating the theory of our framework, as well as showing the tractability of our estimator.[10]

7.2.1 Comparing Two Mechanisms. The (ε, δ) privacy-spectrum generated by our privacy estimator can be used to generate a more in-depth comparison of two mechanisms. For example, suppose that you are presented with two mechanisms, $\mathcal{M}_{L,\varepsilon}$ and $\mathcal{M}_{G,\varepsilon,\delta}$, noised so that they give the privacy guarantees of $(\varepsilon, \delta) = (1, 0)$

[10]Indeed, such a validation is a necessary step to ensure that there is benefit in applying such a method to heuristic algorithms.



(a) Estimated (ε, δ)-privacy spectrum of $\mathcal{M}_{\text{SVT},\varepsilon}$ parameterized with $\varepsilon = 1, \delta = 0$. We see that better ε may be achieved with sacrifices to δ.

(b) Estimated (ε, δ) spectrum of $\mathcal{M}_{\text{SVT},\varepsilon}$ and its two variants. $\mathcal{M}_{\text{SVT2},\varepsilon}$ and $\mathcal{M}_{\text{SVT3},\varepsilon}$ have much worse ε-δ trade-offs and are not $(\varepsilon = 1, \delta = 0)$-DP.

Figure 7: SVT's DP-spectrum in comparison with its two buggy variants.

for $\mathcal{M}_{L,\varepsilon}$ and $(\varepsilon, \delta) = (1, 0.00001)$ for $\mathcal{M}_{G,\varepsilon,\delta}$. It appears then, that $\mathcal{M}_{L,\varepsilon}$ is a strictly better mechanism.

However, the (ε, δ) spectrum of these mechanisms lends to a much better comparison. Our privacy estimator can be provide an estimate (with tight accuracy bounds) of such curves (Figure 6b). While in this $\mathcal{M}_{L,\varepsilon}$ versus $\mathcal{M}_{G,\varepsilon,\delta}$ example, we can actually analytically compute the (ε, δ) spectrum, this may not be possible for all mechanisms. Moreover, even for $\mathcal{M}_{L,\varepsilon}$, there is little information about this curve available, and the theoretical δ given by well-known bounds [26] is loose[11]. Figure 6b shows definitively that in fact $\mathcal{M}_{G,\varepsilon,\delta}$ provides a much stronger DP guarantee most of the time (its δ is closer to 0, even if you claim a smaller ε than 1) while $\mathcal{M}_{L,\varepsilon}$ can only provide $\varepsilon = 1$ DP guarantee but achieves $\varepsilon < 1$ with undesirable δ.

As another application of our framework we plot the estimated privacy spectrum of the SVT mechanism and its two buggy variants (algorithm details in Figure 8, Figure 9 and Figure 10, Appendix H).

Figure 7a plots the privacy parameter ε of $\mathcal{M}_{\text{SVT},\varepsilon=1}$ against its corresponding optimal δ (Def. 4.1). Our estimator verifies that indeed $\mathcal{M}_{\text{SVT},\varepsilon=1}$ provides $(1, 0)$-DP. It also shows $\varepsilon = 1$ is tight, since when a small ε is claimed, Figure 7a demonstrates a significant increase in δ. Figure 7b compares the privacy spectrum of mechanisms $\mathcal{M}_{\text{SVT},\varepsilon}$ and its two variants $\mathcal{M}_{\text{SVT2},\varepsilon}$ and $\mathcal{M}_{\text{SVT3},\varepsilon}$. We see that $\mathcal{M}_{\text{SVT2},\varepsilon}$, $\mathcal{M}_{\text{SVT3},\varepsilon}$ provide much weaker DP guarantee than

[11]For $\mathcal{M}_{G,\varepsilon,\delta}$, because noise distribution's standard deviation is $\sqrt{\mathcal{N}(0, \frac{2\log(1.25/\delta)}{\varepsilon^2})}$, the δ as the function of ε (the top green curve in Figure 6a,) is very loose.

$\mathcal{M}_{\text{SVT},\varepsilon}$ as their corresponding δ is significantly larger for the same ε. Even so, we observe that some reasonable DP guarantee may be provided by $\mathcal{M}_{\text{SVT2},\varepsilon}$, while there is no evidence that $\mathcal{M}_{\text{SVT3},\varepsilon}$ could provide any meaningful DP guarantee. Appendix H) gives a brief explanation of how we estimate these mechanisms using our framework.

7.2.2 Verifying Mechanism Implementation. A common use of privacy estimators has been in verifying (claims about) the privacy of DP mechanisms (e.g., [5, 7]). In Appendix I we show that our estimator is in fact useful also for this task.

ACKNOWLEDGMENT

We would like to thank Maksim Tsikhanovich for initial fruitful discussions on linking measures of privacy to Bayes optimal classification problems. In addition, we thank him for pointing us to his code at [35] which was used in our experiments to implement the empirical bootstrapping method. This piece of code helps us understand the relationship between our estimator's empirical tightness and the number of samples.

REFERENCES

[1] D. Zhang and D. Kifer, "Lightdp: Towards automating differential privacy proofs," in *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, 2017, pp. 888–901.
[2] 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*, 2019, pp. 655–669.
[3] Y. Wang, Z. Ding, D. Kifer, and D. Zhang, "CheckDP: An automated and integrated approach for proving differential privacy or finding precise counterexamples," in *ACM CCS 2020*, J. Ligatti, X. Ou, J. Katz, and G. Vigna, Eds. ACM Press, Nov. 2020, pp. 919–938.
[4] H. Zhang, E. Roth, A. Haeberlen, B. C. Pierce, and A. Roth, "Testing differential privacy with dual interpreters," *Proc. ACM Program. Lang.*, vol. 4, no. OOPSLA, nov 2020.
[5] Z. Ding, Y. Wang, G. Wang, D. Zhang, and D. Kifer, "Detecting violations of differential privacy," in *ACM CCS 2018*, D. Lie, M. Mannan, M. Backes, and X. Wang, Eds. ACM Press, Oct. 2018, pp. 475–489.
[6] B. Bichsel, T. Gehr, D. Drachsler-Cohen, P. Tsankov, and M. T. Vechev, "DP-finder: Finding differential privacy violations by sampling and optimization," in *ACM CCS 2018*, D. Lie, M. Mannan, M. Backes, and X. Wang, Eds. ACM Press, Oct. 2018, pp. 508–524.
[7] B. Bichsel, S. Steffen, I. Bogunovic, and M. T. Vechev, "DP-sniper: Black-box discovery of differential privacy violations using classifiers," in *2021 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2021, pp. 391–409.
[8] X. Liu and S. Oh, "Minimax optimal estimation of approximate differential privacy on neighboring databases," *Advances in neural information processing systems*, vol. 32, 2019.
[9] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 2017, pp. 263–275.
[10] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *arXiv preprint arXiv:1603.01887*, 2016.
[11] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography Conference*. Springer, 2016, pp. 635–658.
[12] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, ser. Stochastic Modelling and Applied Probability. Springer, 1996, vol. 31.
[13] R. Bassily, A. Groce, J. Katz, and A. Smith, "Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy," in *54th FOCS*. IEEE Computer Society Press, Oct. 2013, pp. 439–448.
[14] A. Antos, L. Devroye, and L. Gyorfi, "Lower bounds for bayes error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 643–645, 1999.
[15] Ö. Askin, T. Kutta, and H. Dette, "Statistical quantification of differential privacy: A local approach," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 402–421.
[16] J. Zhang, X. Xiao, and X. Xie, "Privtree: A differentially private algorithm for hierarchical decompositions," in *SIGMOD Conference*. ACM, 2016, pp. 155–170.
[17] M. Lyu, D. Su, and N. Li, "Understanding the sparse vector technique for differential privacy," *Proc. VLDB Endow.*, vol. 10, no. 6, pp. 637–648, 2017.
[18] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: randomized aggregatable privacy-preserving ordinal response," in *CCS*. ACM, 2014, pp. 1054–1067.
[19] T. Humphries, M. Rafuse, L. Tulloch, S. Oya, I. Goldberg, U. Hengartner, and F. Kerschbaum, "Differentially private learning does not bound membership inference," *arXiv preprint arXiv:2010.12112*, 2020.
[20] Ú. Erlingsson, I. Mironov, A. Raghunathan, and S. Song, "That which we call private," *arXiv preprint arXiv:1908.03566*, 2019.
[21] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 2018, pp. 268–282.
[22] K. Chatzikokolakis, G. Cherubin, C. Palamidessi, and C. Troncoso, "The bayes security measure," *arXiv preprint arXiv:2011.03396*, 2020.
[23] B. Yu, *Assouad, Fano, and Le Cam*. New York, NY: Springer New York, 1997, pp. 423–435. [Online]. Available: https://doi.org/10.1007/978-1-4612-1880-7_29
[24] A. C. Gilbert and A. McMillan, "Property testing for differential privacy," in *Allerton*. IEEE, 2018, pp. 249–258.
[25] C. Dwork, "Differential privacy (invited paper)," in *ICALP 2006, Part II*, ser. LNCS, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052. Springer, Heidelberg, Jul. 2006, pp. 1–12.
[26] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, p. 211–407, Aug. 2014. [Online]. Available: https://doi.org/10.1561/0400000042
[27] R. Bhaskar, A. Bhowmick, V. Goyal, S. Laxman, and A. Thakurta, "Noiseless database privacy," in *ASIACRYPT 2011*, ser. LNCS, D. H. Lee and X. Wang, Eds., vol. 7073. Springer, Heidelberg, Dec. 2011, pp. 215–232.
[28] S. Leung and E. Lui, "Bayesian mechanism design with efficiency, privacy, and approximate truthfulness," in *International Workshop on Internet and Network Economics*. Springer, 2012, pp. 58–71.
[29] Y. Duan, "Privacy without noise," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ser. CIKM '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 1517–1520. [Online]. Available: https://doi.org/10.1145/1645953.1646160
[30] L. Ao, Y. Lu, L. Xia, and V. Zikas, "How private are commonly-used voting rules?" in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 629–638.
[31] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.
[32] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM CCS 2016*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM Press, Oct. 2016, pp. 308–318.
[33] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *IEEE Symposium on Security and Privacy*. IEEE, 2019, pp. 332–349.
[34] J. M. Abowd, "The us census bureau adopts differential privacy," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2867–2867.
[35] M. Tsikhanovich, "empirical privacy," https://github.com/maksimt/empirical_privacy, 2019.

A PROOF OF IMPOSSIBILITY OF APPROXIMATE DP ESTIMATOR

Proof of Theorem 4.5. We will prove the theorem by

(1) constructing two mechanisms $\mathcal{M}$ and $\mathcal{M}_D$, where $\mathcal{M}_D$ is a mechanism parameterized with a database D.
(2) showing that there does not exist a polynomial time algorithm P that can distinguish between $\mathcal{M}$ and $\mathcal{M}_D$ if D is randomly chosen.
(3) proving by contradiction that if the algorithm E_ε defined in the lemma exists, then we can turn it into a distinguisher P (which was proven impossible).

We start by constructing two mechanisms $\mathcal{M}$ and $\mathcal{M}_D$. Let $\mathcal{M} : \{0,1\}^n \mapsto \{0,1\}$ and $\mathcal{M}_D : \{0,1\}^n \mapsto \{0,1\}$ be two randomized mechanisms. Let $D \in \{0,1\}^n$. We define $\mathcal{M}$ as the following: no matter what input in the domain it takes, $\mathcal{M}$ outputs 0 with probability $\frac{1}{2}$ otherwise outputs 1 with probability $\frac{1}{2}$. We define $\mathcal{M}_D$ as the following: given any input x not equal to D it outputs $\mathcal{M}(x)$

otherwise $\mathcal{M}_D$ outputs 0 with probability 0 and 1 with probability 1.

We know that $\mathcal{M}$ is $(0,0)$-differential private, because its output is independent of its input. Also, we know that $\mathcal{M}_D$ is $(0,1)$-differential private, because its output is deterministic when given D.

Then, we define the following game for algorithm P: Choose database D uniformly at random from $\{0,1\}^n$. Toss a fair coin b, and give the algorithm P black-box access to either $\mathcal{M}$ or $\mathcal{M}_D$ based on b. The algorithm P wins if it can correctly decide whether it was given $\mathcal{M}$ or $\mathcal{M}_D$.

Since P is running in polynomial time, and has only black-box access to the mechanism, this means we can consider P's output as a randomized function of its poly(n) queries $D_1, D_2, \cdots$ (made possibly adaptively) to the mechanism. Since $\mathcal{M}$'s and $\mathcal{M}_D$'s outputs only differ on input D, and D is chosen uniformly at random, it means the probability that P queries D is negligible in n. In other words, P can only win with at best negligibly better probability than guessing $(1/2)$.

We now prove by contradiction that E_ε defined in the lemma does not exist. Suppose for contradiction that E_ε does indeed exist. Then, let P do the following: given a mechanism (one of $\mathcal{M}$ or $\mathcal{M}_D$), feed this mechanism and $\varepsilon = 0$ to E_ε. If E_ε says an estimate $\delta' \leqslant \alpha$, P guesses that it was given $\mathcal{M}$. Else, it guesses that it was given $\mathcal{M}_D$. Since, with probability $\frac{1}{2} + \nu(n)$, E_ε should always give some estimate $\delta' \in [0, \alpha]$ given $\mathcal{M}$, and some estimate $\delta' \in [1-\alpha, 1]$ given $\mathcal{M}_D$, it means P should be correct with probability at least $\frac{1}{2} + \nu(n)$. This contradicts the conclusion of (2), meaning E_ε does not exist. □

B PROOF OF PROPERTIES OF RELATIVE DP

Proof of Prop. 4.7. This proposition holds by definition of differential privacy. □

Proof of Prop. 4.8. By the relative DP definition and the proposition's condition, the mechanism $\mathcal{M}$ satisfies that, for every neighboring databases $D \simeq D'$ and $D' \simeq D : D \in \mathcal{T}$ and subset $\mathcal{S} \subseteq \mathsf{Range}(\mathcal{M})$,

$$\begin{aligned}\Pr[\mathcal{M}(D) \in \mathcal{S}] &\leqslant e^{\varepsilon_i} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta_i \\ &\leqslant e^{\max_{i\in[k]} \varepsilon_i} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \max_{i\in[k]} \delta_i,\end{aligned}$$

which completes the proof. □

Proof of Prop. 4.9. Let $D = (D_1, \cdots, D_k)$ be a arbitrary database from the set $\mathcal{T}_1 \times \cdots \times \mathcal{T}_k$. Let $D' = (D'_1, \cdots, D'_k)$ be a arbitrary neighbor of D. Without loss of generality, let D have an extra record x than D' in the j-th partition, that is $D_j = D'_j \bigcup \{x\}$, otherwise $D_i = D'_i$ for $i \in [k]$ and $i \neq j$. For every subset $\mathcal{S} \subseteq \mathsf{Range}(\mathcal{M})$, we have

$$\begin{aligned}&\Pr[\mathcal{M}(D) \in \mathcal{S}] \\ &= \Pr[(\mathcal{M}_1(D_1), \cdots, \mathcal{M}_k(D_k)) \in (\mathcal{S}_1, \cdots, \mathcal{S}_k)] \\ &= \prod_{i\in[k]} \Pr[\mathcal{M}_i(D_i) \in \mathcal{S}_i] \\ &= \Pr[\mathcal{M}_j(D_j) \in \mathcal{S}_j] \prod_{i\in[k]\setminus\{j\}} \Pr[\mathcal{M}_i(D_i) \in \mathcal{S}_i] \\ &\leqslant (e^{\varepsilon_j} \Pr[\mathcal{M}_j(D'_j) \in \mathcal{S}_j] + \delta_j) \prod_{i\in[k]\setminus\{j\}} \Pr[\mathcal{M}_i(D'_i) \in \mathcal{S}_i] \\ &\leqslant e^{\varepsilon_j} \Pr[\mathcal{M}_j(D'_j) \in \mathcal{S}_j] \prod_{i\in[k]\setminus\{j\}} \Pr[\mathcal{M}_i(D'_i) \in \mathcal{S}_i] + \delta_j \\ &= e^{\varepsilon_j} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta_j \\ &\leqslant (\max_{i\in[k]} e^{\varepsilon_i}) \Pr[\mathcal{M}(D') \in \mathcal{S}] + (\max_{i\in[k]} \delta_i),\end{aligned}$$

which completes the proof. □

Proof of Prop. 4.10. Let D be a arbitrary database from the set $\mathcal{T}$ and D' be a arbitrary neighbor of D, that is, $D' \simeq D$ or $D \simeq D'$. For every subset $\mathcal{S} \subseteq \mathsf{Range}(\mathcal{M})$, we have

$$\begin{aligned}&\Pr[\mathcal{M}(D) \in \mathcal{S}] \\ &= \Pr[(\mathcal{M}_1(D), \cdots, \mathcal{M}_k(D)) \in (\mathcal{S}_1, \cdots, \mathcal{S}_k)] \\ &= \prod_{i\in[k]} \Pr[\mathcal{M}_i(D) \in \mathcal{S}_i] \\ &= \prod_{i\in[k-1]} \Pr[\mathcal{M}_i(D) \in \mathcal{S}_i]) \Pr[\mathcal{M}_k(D) \in \mathcal{S}_k] \\ &\leqslant \prod_{i\in[k-1]} \Pr[\mathcal{M}_i(D) \in \mathcal{S}_i] \left(e^{\varepsilon_k} \Pr[\mathcal{M}_k(D') \in \mathcal{S}_k] + \delta_k\right) \\ &\leqslant e^{\varepsilon_k} \Big(\prod_{i\in[k-1]} \Pr[\mathcal{M}_i(D) \in \mathcal{S}_i]) \Pr[\mathcal{M}_k(D') \in \mathcal{S}_k] + \delta_k \\ &\leqslant e^{\sum_{i\in[k]} \varepsilon_i} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \sum_{i\in[k]} \delta_i,\end{aligned}$$

which completes the proof. □

Proof of Prop. 4.11. Let D be a arbitrary database from the set $\mathcal{T}$ and D' be a arbitrary neighbor of D, that is, $D' \simeq D$ or $D \simeq D'$. For every subset $\mathcal{S} \subseteq \mathsf{Range}(\mathcal{M})$, define set $T = \{t \in \mathsf{Range}(\mathcal{M}_1) : f(t) \in \mathcal{S}\}$. We have

$$\begin{aligned}\Pr[\mathcal{M}(D) \in \mathcal{S}] &= \Pr[f(\mathcal{M}_1(D)) \in \mathcal{S}] \\ &= \sum_{t\in T} \Pr[\mathcal{M}_1(D) = t] \\ &= \Pr[\mathcal{M}_1(D) \in T] \\ &\leqslant e^{\varepsilon} \Pr[\mathcal{M}_1(D') \in T] + \delta, \\ &= e^{\varepsilon} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta.\end{aligned}$$

which completes the proof. □

C PROOF: CONNECTING δ IN (ε, δ)-DP WITH RISK OF BAYES CLASSIFIER

Proof of Theorem 5.4. Let $\Delta\left(\left[\mathcal{M}(D)\right]_\varepsilon, \mathcal{M}(D')\right)$ be the statistical distance between $\left[\mathcal{M}(D)\right]_\varepsilon$ and $\mathcal{M}(D')$. Our plan of proof is the following. We first show the equivalence between the optimal $\delta_{D,D'}$ and the statistical distance $\Delta\left(\left[\mathcal{M}(D)\right]_\varepsilon, \mathcal{M}(D')\right)$.

Claim 1. *The following equation between the optimal $\delta_{D,D'}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$ and the statistical distance $\Delta\left(\left[\mathcal{M}(D)\right]_\varepsilon, \mathcal{M}(D')\right)$ holds:*

$$\delta_{D,D'} = \max\left(e^{\varepsilon}\left(\Delta\left(\left[\mathcal{M}(D)\right]_\varepsilon, \mathcal{M}(D')\right) - (1 - e^{-\varepsilon})\right), 0\right).$$

Proof of Claim 1. By definition of optimal $\delta_{D,D'}$ in Definition 4.1, we have

$$\begin{aligned}\delta_{D,D'} &= \max\Big(\max_{S\subseteq O}\Pr[\mathcal{M}(D)\in\mathcal{S}] - e^{\varepsilon}\Pr[\mathcal{M}(D')\in\mathcal{S}], 0\Big)\\ &= \max\Big(e^{\varepsilon}\max_{S\subseteq O}\Big(e^{-\varepsilon}\Pr[\mathcal{M}(D)\in S] - \Pr[\mathcal{M}(D')\in S]\Big), 0\Big).\end{aligned} \tag{1}$$

We first check that the distribution $\left[\mathcal{M}(D)\right]_{\varepsilon}$ has the following property, for all $\mathcal{S}\in O$ (support of mechanism $\mathcal{M}$),

$$\Pr\left[\left[\mathcal{M}(D)\right]_{\varepsilon}\in\mathcal{S}\right] = e^{-\varepsilon}\Pr[\mathcal{M}(D)\in\mathcal{S}].$$

This is because, for all $\mathcal{S}\in O$,

$$\begin{aligned}\Pr[\left[\mathcal{M}(D)\right]_{\varepsilon}\in\mathcal{S}] &= \Pr[c=1\wedge\mathcal{M}(D)\in\mathcal{S}]\\ &= \Pr[c=1]\Pr[\mathcal{M}(D)\in\mathcal{S}] \qquad (c \text{ and } \mathcal{M}(D) \text{ are independent})\\ &= e^{-\varepsilon}\Pr[\mathcal{M}(D)\in\mathcal{S}].\end{aligned}$$

We are given a method to find the statistical distance between two distributions by sampling them. The statistical distance between distributions $\left[\mathcal{M}(D)\right]_{\varepsilon}$ and $\mathcal{M}(D')$ is defined as follows:

$$\begin{aligned}&\Delta\Big(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\Big)\\ &\equiv \max_{S\subseteq O}\Big(\Pr\Big[\left[\mathcal{M}(D)\right]_{\varepsilon}\in\mathcal{S}\Big] - \Pr[\mathcal{M}(D')\in\mathcal{S}]\Big).\end{aligned}$$

By construction, $\left[\mathcal{M}(D)\right]_{\varepsilon}$ outputs $\perp$ with probability $1-e^{-\varepsilon}$, whereas $\mathcal{M}(D')$ outputs $\perp$ with probability zero. Thus, $\perp$ can always be included in the set that maximizes the statistical distance.

$$\begin{aligned}&\Delta(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D'))\\ &= \max_{S\in O}\Big(\Pr[\left[\mathcal{M}(D)\right]_{\varepsilon}\in\mathcal{S}] - \Pr[\mathcal{M}(D')\in\mathcal{S}]\Big)\\ &+ \Big((\Pr[\left[\mathcal{M}(D)\right]_{\varepsilon}=\perp] - \Pr[\mathcal{M}(D')=\perp]\Big)\\ &= \max_{S\in O}\Big(e^{-\varepsilon}\Pr[\mathcal{M}(D)\in\mathcal{S}] - \Pr(\mathcal{M}(D')\in\mathcal{S})\Big) + (1-e^{-\varepsilon})\end{aligned}$$

Then, plug the above equation into the equation 1, we have

$$\delta_{D,D'} = \max(e^{\varepsilon}\Big(\Delta\Big(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\Big) - (1-e^{-\varepsilon})\Big), 0),$$

which completes the proof. □

Secondly, we show the equivalence between risk of the the Bayes classifier $R(h^*_{D,D'})$ and the statistical distance $\Delta\Big(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\Big)$.

Claim 2.

$$\Delta\Big(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\Big) = 2\cdot\left(\frac{1}{2} - R(h^*_{D,D'})\right).$$

Proof of Claim 2. The statistical distance can be alternatively defined as

$$\begin{aligned}&\Delta\Big(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\Big)\\ &= \max_{h}\Big|\Pr_{x\sim\mathcal{M}(D')}[h(x)=1] - \Pr_{x\sim\left[\mathcal{M}(D)\right]_{\varepsilon}}[h(x)=1]\Big|,\end{aligned}$$

where h is any classifier for the distribution $\mathcal{P}$. Then,

$$\begin{aligned}&\Delta\Big(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\Big)\\ &= 2\left(\frac{1}{2}\max_{h}\Big|\Pr_{x\sim\mathcal{M}(D')}[h(x)=1] - \Big(1-\Pr_{x\sim\left[\mathcal{M}(D)\right]_{\varepsilon}}[h(x)=0]\Big)\Big|\right)\\ &= 2\left(\max_{h}\Big|\frac{1}{2}\Big(\Pr_{x\sim\mathcal{M}(D')}[h(x)=1] + \Pr_{x\sim\left[\mathcal{M}(D)\right]_{\varepsilon}}[h(x)=0]\Big) - \frac{1}{2}\Big|\right)\\ &= 2\Big(\max_{h}\Big|\Pr_{(x,y)\sim\mathcal{P}}[h(x)=1, y=1] + \Pr_{(x,y)\sim\mathcal{P}}[h(x)=0, y=0] - \frac{1}{2}\Big|\Big)\\ &= 2\Big(\max_{h}\Big|\Pr_{(x,y)\sim\mathcal{P}}[h(x)=y] - \frac{1}{2}\Big|\Big)\\ &= 2\Big(\max_{h}\Big|1 - \Pr_{(x,y)\sim\mathcal{P}}[h(x)\neq y] - \frac{1}{2}\Big|\Big)\\ &= 2\Big(\max_{h}\Big|\frac{1}{2} - R(h)\Big|\Big)\\ &= 2\Big(\frac{1}{2} - R(h^*_{D,D'})\Big).\end{aligned}$$

□

Show the equivalence between the optimal $\delta_{D,D'}$ and the risk of the the Bayes classifier $R(h^*)$. Combining the Claim 1 and the Claim 2, it is easy to show that

$$\delta_{D,D'} = \max\Big(1 - 2e^{\varepsilon}R(h^*_{D,D'}), 0\Big),$$

which completes the proof. □

D PROOF: GENERAL ESTIMATOR

Proof of Lemma 5.6. For every $(\mathcal{M}, D, D', \varepsilon)$, and its corresponding distribution $\mathcal{P}$, we have the following. Recall the random variable r as computed in Step 4, Figure 1, is the testing risk for classifier $h^{\mathsf{B}}_{n_1}$ with n_2 testing samples. We could show that r is a good approximate of the risk of the Bayes classifier $R(h^*_{D,D'})$.

Claim 3. *With probability at least* $1-\beta$,

$$|r - R(h^*_{D,D'})| = O\left(g(\mathcal{X}, n/2, \beta/2) + \sqrt{\ln(1/\beta)/n}\right).$$

Proof of Claim 3. Recall $n_1 = n/2$, defined in Step 1, Fig. 1. By the condition in the Lemma, when the sample size parameter n_1 is large enough, we have that, with probability at least $1-\beta/2$,

$$|R(h^{\mathsf{B}}_{n_1}) - R(h^*_{D,D'})| \leqslant c\cdot g(\mathcal{X}, n_1, \beta/2) = c\cdot g(\mathcal{X}, n/2, \beta/2),$$

where c is a constant.

By Theorem 3.7, plug in $n_2 = n/2$ (defined in Step 1, Fig. 1), with probability at least $1-\beta/2$, we have

$$|r - R(h^{\mathsf{B}}_{n_1})| \leqslant \sqrt{\ln(4/\beta)/n}.$$

Apply union bound and triangular inequality to above two inequalities with probability at least $1-\beta$, we have

$$\begin{aligned}|r - R(h^*)| &\leqslant |r - R(h^{\mathsf{B}}_{n_1})| + |R(h^{\mathsf{B}}_{n_1}) - R(h^*_{D,D'})|\\ &\leqslant c\cdot g(\mathcal{X}, n/2, \beta/2) + \sqrt{\ln(4/\beta)/n},\end{aligned}$$

which completes the proof. □

Using Claim 3, we could show that $\delta'_{D,D'}$ (defined in Step 5, Fig. 1) is a good approximate of $\delta_{D,D'}$ with respect to $(\mathcal{M}, D, D', \varepsilon)$.

Claim 4. *With probability at least* $1-\beta$,

$$|\delta'_{D,D'} - \delta_{D,D'}| = O\left(g(\mathcal{X}, n/2, \beta/2) + \sqrt{\ln(1/\beta)/n}\right).$$

Proof of Claim 4.

$$
\begin{aligned}
&\left|\delta'_{D,D'} - \delta_{D,D'}\right| \\
&= \left|\max\left(1 - 2e^{\varepsilon} r, 0\right) - \delta_{D,D'}\right| && \text{(By Fig. 1, Step 5,)} \\
&= \left|\max\left(1 - 2e^{\varepsilon} r, 0\right) - \max\left(1 - 2e^{\varepsilon} R\left(h^*_{D.D'}\right), 0\right)\right| && \text{(By Theorem 5.4)} \\
&\leqslant \left|\left(\left(1 - 2e^{\varepsilon} r\right) - \left(1 - 2e^{\varepsilon} R\left(h^*_{D.D'}\right)\right)\right)\right| \\
&\leqslant 2e^{\varepsilon}\left|r - R(h^*_{D.D'})\right| \\
&= O\left(g(\mathcal{X}, n/2, \beta/2) + \sqrt{\ln(1/\beta)/n}\right), && \text{(By Claim 3)}
\end{aligned}
$$

where the last step we omit the constant $2e^{\epsilon}$ since the tight bound is in asymptotic form. □

Combining the results of Claim 3 and Claim 4, we have that for every tuple $(\mathcal{M}, D, D', \varepsilon)$ the algorithm $\mathcal{A}^{\mathsf{B}}_{C}$ provides a $\alpha = O\left(g(\mathcal{X}, n/2, \beta/2) + \sqrt{\ln(1\beta)/n}\right)$ tight bound with probability $1 - \beta$. Thus concludes the proof that $\mathcal{A}^{\mathsf{B}}_{C}$ is a (α, β)-Approximate δ-Estimator for a Pair of Databases for C. □

E PROOF: ESTIMATOR USING KNN

Proof of Theorem 5.7. The algorithm $\mathcal{A}^{\mathsf{kNN}}_{C}$ with the classification algorithm kNN is a concrete instantiation of $\mathcal{A}^{\mathsf{B}}_{C}$, shown in Figure 1. To prove that $\mathcal{A}^{\mathsf{kNN}}_{C}$ is a (α, β)-Approximate δ-Estimator for a Pair of Databases for C, we could directly plug in the convergence results of kNN into Lemma 5.6 and then complete the proof.

For every tuple $(\mathcal{M}, D, D', \varepsilon)$, where $\mathcal{M} \in C$, we have two random variables: $\mathcal{M}(D')$ and $\left[\mathcal{M}(D)\right]_{\varepsilon}$. We also have a corresponding distribution $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ (Def. 5.3, abbreviated below as $\mathcal{P}$). Recall that the experiment of generating $\mathcal{P}$ is following: Toss a fair coin b. If $b = 0$ the experiment outputs a sample o according to distribution $\left[\mathcal{M}(D)\right]_{\varepsilon}$, or otherwise outputs a sample o according to distribution $\mathcal{M}(D')$.

Let h^* and $R(h^*)$ be the Bayes classifier and the risk of the Bayes classifier for the distribution $\mathcal{P}$, respectively. Step 3 of algorithm $\mathcal{A}^{\mathsf{kNN}}_{C}$ (Figure 1) computes a kNN classifier h^{NN}_{k,n_1} for distribution $\mathcal{P}$. Step 4 computes $\hat{R}_{n_2}(h^{\mathsf{NN}}_{k,n_1})$, the testing risk of h^{NN}_{k,n_1} with n_2 testing samples.

Because $\mathcal{M} \in C$, the distribution of $\mathcal{M}(D')$ has density. Moreover, the distribution $\left[\mathcal{M}(D)\right]_{\varepsilon}$ *almost* has a density except at point $\perp$. By Chapter 11.2 of [12], the density assumption was needed to avoid problems caused by training points having equal distances to testing points (i.e., so that each point has exactly *k-nearest neighbors*). For the point $\perp$, we could define the distance from it to any other points as infinity, so at point $\perp$ the distance tie problem does not appear even without the density assumption. This means we could still use the result from Theorem 3.8. Thus, Theorem 3.8's condition suffices. By Theorem 3.8, when the sample size parameter n_1 is large enough, we have that

$$\Pr[\,|R(h^{\mathsf{NN}}_{k,n_1}) - R(h^*)| > \alpha] \leqslant 2e^{-n_1\alpha^2/(72c_d^2)}.$$

Recall $n_1 = n/2$, defined in Step 1, Fig. 1. Set $2e^{-n_1\alpha^2/(72c_d^2)} = \beta/2$. Rearranging the inequality, with probability at least $1 - \beta/2$,

$$|R(h^{\mathsf{NN}}_{k,n_1}) - R(h^*)| \leqslant 12c_d\sqrt{\ln(4\beta)n} \tag{2}$$

Plug the above inequality into Lemma 5.6, we have that for every $\delta_{D,D'}$ with respect to the $(\mathcal{M}, D, D', \varepsilon)$ and its estimate $\delta'_{D,D'}$ (defined in Step 5, Fig. 1)

$$|\delta'_{D,D'} - \delta_{D,D'}| \leqslant 12c_d\sqrt{\ln(4\beta)n} + O\left(\sqrt{\ln(1/\beta)/n}\right),$$

which completes the proof. □

F PROOF: RELATIVE-DP ESTIMATOR USING KNN

Proof of Theorem 5.10. Let q be the number of neighboring databases $D \simeq D'$ where $D \in \mathcal{T}$. Let $\{\delta_1, \cdots, \delta_{2q}\}$ be the set of optimal $\delta_{D,D'}$ (and $\delta_{D',D}$) for each neighboring databases, $\{\delta'_1, \cdots, \delta'_{2q}\}$ (computed in Step 1, Fig. 2) be the set of estimate for $\{\delta_1, \cdots, \delta_{2q}\}$. δ'_1 is the estimate of δ_1, etc.

By Theorem 5.7, we could say that for each $i \in [2q]$, with probability at least $1 - \beta/2q$, for a constant c

$$|\delta'_i - \delta_i| \leqslant c \cdot c_d\sqrt{\ln(2q/\beta)/n},$$

By a union bound, with probability at least $1 - \beta$,

$$\max_{i\in[2q]} |\delta'_i - \delta_i| \leqslant c \cdot c_d\sqrt{\ln(2q/\beta)/n}. \tag{3}$$

Denote the index of $\delta_{\mathcal{T}}$ in set $\{\delta_1, \cdots, \delta_{2q}\}$ as a. That is $\delta_{\mathcal{T}} = \delta_a = \max_{i\in[2q]} \delta_i$. Denote the index of the maximum estimate in set $\{\delta'_1, \cdots, \delta'_{2q}\}$ as b. That is $\delta'_b = \max_{i\in[2q]} \delta'_i$. The algorithm $\mathcal{A}^{B}_{C,t}$ outputs δ'_b as the estimate of $\delta_{\mathcal{T}}$. Then, with probability at least $1 - \beta$,

$$
\begin{aligned}
|\delta'_b - \delta_{\mathcal{T}}| &= |\delta'_b - \delta_a| \\
&\leqslant \max\left(|\delta'_b - \delta_b|, |\delta'_a - \delta_a|\right) \\
&\leqslant \max_{i\in[2q]} |\delta'_i - \delta_i|
\end{aligned} \tag{4}
$$

We bound the total number of neighboring databases q. Because the size of the databases set $\mathcal{T}$ is smaller than t and each databases has at most m records, hence by Definition 3.2 each database has at most m neighbors, so that

$$q \leqslant tm. \tag{5}$$

Combining Inequalities 3, 4 and 5, with probability at least $1 - \beta$,

$$|\delta'_b - \delta_{\mathcal{T}}| \leqslant c \cdot c_d\sqrt{\ln(2tm/\beta)/n},$$

which completes the proof. □

G ANALYTICAL COMPUTED PRIVACY OF LAPLACIAN AND GAUSSIAN MECHANISM

Definition G.1 (The Laplacian bit query mechanism $\mathcal{M}_{L,\varepsilon}$). Let $\mathcal{M}_{L,\varepsilon}$ denote the differentially private bit query mechanism using Laplacian mechanism, which takes a bit b as input, samples a noise $v \sim Lap(\varepsilon)$ according to Laplace distribution[12], and then returns $b+v$ as the mechanism's output. $\mathcal{M}_{L,\varepsilon}$ is $(\varepsilon, 0)$-differential private [25].

Definition G.2 (The Gaussian bit query mechanism $\mathcal{M}_{G,\varepsilon,\delta}$). Let $\mathcal{M}_{G,\varepsilon,\delta}$ denote the differentially private bit query mechanism using Gaussian mechanism, which takes a bit b as input, samples a noise

[12]The Laplace distribution (centered at 0) with parameter λ is the distribution with probability density function: $Lap(x \mid \lambda) = \frac{\lambda}{2}\exp\left(-\lambda|x|\right)$. We use $Lap(\lambda)$ to denote the Laplace distribution with parameter λ.

$v \sim \mathcal{N}(0, 2\varepsilon^{-2}\log(1.25/\delta))$ according to Gaussian distribution[13], and then returns $b + v$ as the mechanism's output. $\mathcal{M}_{G,\varepsilon,\delta}$ is (ε, δ)-differential private [25].

Lemma G.3. *Let $\mathcal{M}_{L,\varepsilon}$ be the noised bit query mechanism defined in Definition G.1. Let $\delta(\varepsilon')$ be the optimal δ (Def. 4.1) with respect to the tuple $(\mathcal{M}_{L,\varepsilon}, \varepsilon')$. $\delta(\varepsilon')$ satisfies the following equality*

$$\delta'(\varepsilon') = \begin{cases} 1 - e^{-\frac{1}{2}(\varepsilon - \varepsilon')} & \varepsilon' \in [0, \varepsilon] \\ 0 & \varepsilon' \geqslant \varepsilon. \end{cases} \tag{6}$$

Proof. Note that $\mathcal{M}_{L,\varepsilon}$ has only one neighboring database pair $(D, D') = (0, 1)$. By Definition 4.1, we have

$$\delta(\varepsilon') = \max(\max_{S \subseteq O} \Pr[\mathcal{M}_{L,\varepsilon}(D) \in S] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') \in S], 0),$$

where $O = \mathsf{Range}(\mathcal{M}_{L,\varepsilon})$.

For $\varepsilon' \geqslant \varepsilon$, by the differential privacy definition shown in Definition 3.3, we know

$$\max_{S \subseteq O} \Pr[\mathcal{M}_{L,\varepsilon}(D) \in S] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') \in S] \leqslant 0,$$

so that

$$\delta(\varepsilon') = 0.$$

Now we turn to the case $\varepsilon' < \varepsilon$. We first recall the probability density function of $\mathcal{M}_{L,\varepsilon}(D)$

$$\Pr[\mathcal{M}_{L,\varepsilon}(D) = x] = \frac{\varepsilon}{2} e^{-\varepsilon|x|},$$

where $x \in \mathsf{R}$. Similarly, the probability density function of $\mathcal{M}_{L,\varepsilon}(D')$ is

$$\Pr[\mathcal{M}_{L,\varepsilon}(D') = x] = \frac{\varepsilon}{2} e^{-\varepsilon|x-1|},$$

where $x \in \mathsf{R}$.

For $\varepsilon' < \varepsilon$,

$$\begin{aligned} \delta(\varepsilon') &= \max(\max_{S \subseteq O} \Pr[\mathcal{M}_{L,\varepsilon}(D) \in S] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') \in S], 0) \\ &= \max_{S \subseteq O} \Pr[\mathcal{M}_{L,\varepsilon}(D) \in S] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') \in S] \\ &= \int_{-\infty}^{\infty} \max(0, \Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x]) dx \end{aligned} \tag{7}$$

Denote $x_+ \in \mathsf{R}$ such that $e^{-\varepsilon|x_+|} - e^{\varepsilon'} e^{-\varepsilon|x_+ - 1|} = 0$. The function $\Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x]$ has only one zero, that is x_+. For all $x \leqslant x_+$, $\Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x] \geqslant 0$, otherwise $\Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x] < 0$. One can show

$$x_+ = \frac{1}{2}(1 - \frac{\varepsilon'}{\varepsilon}).$$

Plug in the equation 7, we have

$$\begin{aligned} \delta(\varepsilon') &= \int_{-\infty}^{x_+} \Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x] dx \\ &= \int_{-\infty}^{x_+} \frac{\varepsilon}{2} (e^{-\varepsilon|x|} - e^{\varepsilon'} e^{-\varepsilon|x-1|}) dx \\ &= 1 - e^{-\frac{1}{2}(\varepsilon - \varepsilon')}, \end{aligned}$$

where the last step is by integration. □

[13]The Gaussian distribution with expectation 0 and variance σ^2 is the distribution with probability density function: $\mathcal{N}(x|\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{x^2}{2\sigma^2})$.We use $\mathcal{N}(0, \sigma^2)$ to denote the Gaussian distribution with expectation 0 and variance σ^2

Lemma G.4. *Let $\mathcal{M}_{G,\varepsilon,\delta}$ be the noised bit query mechanism defined in Definition G.2. Let $\delta(\varepsilon')$ be the optimal δ (defined in Def. 4.1) with respect to the tuple $(\mathcal{M}_{G,\varepsilon,\delta}, \varepsilon')$. $\delta(\varepsilon')$ satisfies the following equality*

$$\delta(\varepsilon') = \frac{1}{2}[1 + erf(\frac{x_+}{\sigma\sqrt{2}}) - e^{\varepsilon'}(1 + erf(\frac{x_+ - 1}{\sigma\sqrt{2}}))],$$

where $\sigma^2 = \frac{2\log(1.25/\delta)}{\varepsilon^2}$, $\varepsilon' > 0$, $x_+ = \frac{1}{2}(1 - 2\sigma^2\varepsilon')$ and $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-s^2} ds$ (the standard error function.)

Proof. Note that $\mathcal{M}_{G,\varepsilon,\delta}$ has only one neighboring database pair $(D, D') = (0, 1)$. By Definition 4.1, we have

$$\delta(\varepsilon') = \max(\max_{S \subseteq O} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) \in S] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') \in S], 0),$$

where $O = \mathsf{Range}(\mathcal{M}_{G,\varepsilon,\delta})$.

We then recall the probability density function of $\mathcal{M}_{G,\varepsilon,\delta}(D)$

$$\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}},$$

where $x \in \mathsf{R}$. Similarly, the probability density function of $\mathcal{M}_{G,\varepsilon,\delta}(D')$ is

$$\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-1)^2}{2\sigma^2}},$$

where $x \in \mathsf{R}$.

$x_+ = \frac{1}{2}(1 - 2\sigma^2\varepsilon')$ is the value such that $\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x_+] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x_+] = 0$. The function $\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x]$ has only one zero, that is x_+. For all $x \leqslant x_+$, $\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] \geqslant 0$, otherwise $\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] < 0$.

Now we have, for all $\varepsilon' > 0$,

$$\begin{aligned} \delta(\varepsilon') &= \max(\max_{S \subseteq O} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) \in S] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') \in S], 0) \\ &= \max_{S \subseteq O} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) \in S] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') \in S] \\ &= \int_{-\infty}^{\infty} \max(0, \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x]) dx \\ &= \int_{-\infty}^{x_+} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] dx \\ &= \int_{-\infty}^{x_+} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \int_{-\infty}^{x_+} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] \\ &= (\frac{1}{2} + \frac{1}{2} erf(\frac{x_+}{\sigma\sqrt{2}})) - e^{\varepsilon'}(\frac{1}{2} + \frac{1}{2} erf(\frac{x_+ - 1}{\sigma\sqrt{2}})) \\ &= \frac{1}{2}[1 + erf(\frac{x_+}{\sigma\sqrt{2}}) - e^{\varepsilon'}(1 + erf(\frac{x_+ - 1}{\sigma\sqrt{2}}))], \end{aligned}$$

which completes the proof. □

H ESTIMATING SVT'S PRIVACY SPECTRUM

In this section, we further discuss our SVT experiments on the $\mathcal{M}_{\mathsf{SVT},\varepsilon}$, $\mathcal{M}_{\mathsf{SVT2},\varepsilon}$, $\mathcal{M}_{\mathsf{SVT3},\varepsilon}$ mechanisms. First, to estimate the optimal δ (Def 4.1), we use the link between differential privacy and Bayes optimal risk established in Theorem 5.4. Here, we estimate the Bayes optimal risk for SVT by computing its output on at most some finite k queries. In our experiments, we use $k = 40$, and for simplicity consider integer-output queries and thresholds that are no more than 2 away from the true query output. Lastly, we further reduce the number of samples required by our algorithm by observing that SVT's output distribution is the same on databases D_1

Input: A database D, a counting query list $\mathcal{Q} = \{q_1, q_2, \cdots\} \in \mathcal{N}^*_{\geqslant 0}$, a threshold list $\mathcal{T} = \{T_1, T_2, \cdots\} \in \mathcal{N}^*$.
Output: A bits sequence $s \in \{1, 01, 001, \cdots\}$.

(1) $\rho = Lap\left(\frac{\varepsilon}{2}\right)$
(2) For each query $q_i \in \mathcal{Q}$:
  (a) $\nu_i = Lap\left(\frac{\varepsilon}{4}\right)$
  (b) If $q_i(D) + \nu_i \geqslant T_i + \rho$ then
    (i) Output $a_i = 1$ and **Abort**.
  (c) Else Output $a_i = 0$.

Figure 8: The SVT (Sparse Vector Technique) mechanism $\mathcal{M}_{\text{SVT},\varepsilon}$ (Alg.1 from [17])

Input: A database D, a counting query list $\mathcal{Q} = \{q_1, q_2, \cdots\} \in \mathcal{N}^*_{\geqslant 0}$, a threshold list $\mathcal{T} = \{T_1, T_2, \cdots\} \in \mathcal{N}^*$.
Output: A bits sequence $s \in \{1, 01, 001, \cdots\}$.

(1) $\rho = Lap\left(\frac{\varepsilon}{4}\right)$
(2) For each query $q_i \in \mathcal{Q}$:
  (a) $\nu_i = Lap\left(\frac{3\varepsilon}{4}\right)$
  (b) If $q_i(D) + \nu_i \geqslant T_i + \rho$ then
    (i) Output $a_i = 1$ and **Abort**
  (c) Else Output $a_i = 0$.

Figure 9: A buggy variant of the SVT mechanism $\mathcal{M}_{\text{SVT2},\varepsilon}$ (Alg.4 from [17])

Input: A database D, a counting query list $\mathcal{Q} = \{q_1, q_2, \cdots\} \in \mathcal{N}^*_{\geqslant 0}$, a threshold list $\mathcal{T} = \{T_1, T_2, \cdots\} \in \mathcal{N}^*$.
Output: A bits sequence $s \in \{0, 1\}^*$.

(1) $\rho = Lap\left(\frac{\varepsilon}{2}\right)$
(2) For each query $q_i \in \mathcal{Q}$:
  (a) If $q_i(D) \geqslant T_i + \rho$ then Output $a_i = 1$.
  (b) Else Output $a_i = 0$.

Figure 10: A buggy variant of the SVT mechanism $\mathcal{M}_{\text{SVT3},\varepsilon}$ (Alg.5 from [17])

and D_2, if $q_i(D_1) - T_i = q_i(D_2) - T_i$. Thus, it suffices to test fewer number of databases. For more detail, please see our full version.

I VERIFYING MECHANISM IMPLEMENTATION

Perhaps a more common application of our privacy estimator is to verify the correctness of a mechanism implementation—that is, whether a mechanism implementation really is (ε, δ)-DP as claimed. Compared with previous work, our estimator has the advantage of only requiring black box access to the mechanism, and generating outputs with tight accuracy bounds. Moreover, our estimator can handle even mechanisms with large output spaces. In Fig. 4b, we demonstrate an example of checking whether a mechanism satisfies $(\varepsilon = 1, \delta = 0)$-(relative) DP, by testing the mechanism on $\varepsilon = 1$ and receiving the estimated optimal δ—in this example, δ is a small value on the order of 10^{-5}. This tells us that the true ε is likely close if not equal to 1, when $\delta = 0$. For 2^{26} testing/training samples (or about 10 minutes running time on our machine, a Dell compute node with two 64-core AMD Epyc 7662 "Rome" processors and 256 GB memory), we get an error for δ of around 0.0001, which can be improved by increasing the number of samples. If the privacy spectrum is actually known for this mechanism (which is

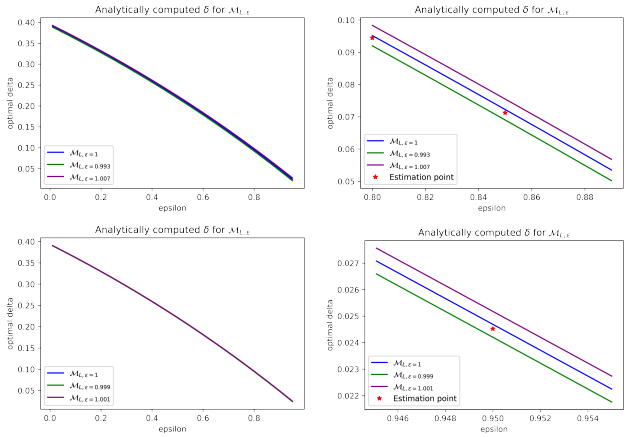


Figure 11: Application 2: verify implementation of $\mathcal{M}_{L,\varepsilon}$ mechanism, by checking which ε, δ trade-off curve the implementation falls under. Different curves represent $\mathcal{M}_{L,\varepsilon}$ with different amount of added noise.

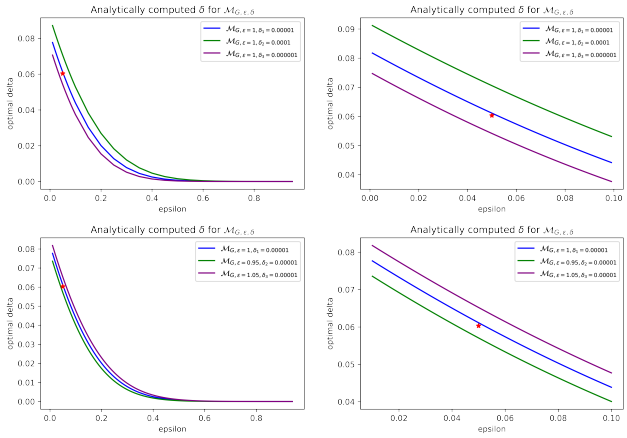


Figure 12: Application 2: verify the mechanism $\mathcal{M}_{G,\varepsilon,\delta}(\varepsilon = 1, \delta = 0.00001)$ is correctly implemented

the case for Laplace and Gaussian mechanisms, via Lemmas G.3 and G.4), then our verification can be even more accurate. To do so, we first generate several analytically computed (ε, δ) curves for $\mathcal{M}_{L,\varepsilon}$, w.r.t. added noise that guarantees at least $(\varepsilon, \delta = 0)$-DP, for $\varepsilon = 0.999, 1, 1.001$. We see (Fig. 11) that the ε, δ trade-off of the implementation is the closest to the analytically computed curve generated by mechanism $\mathcal{M}_{L,\varepsilon}$ with noise according to $\varepsilon = 1$, which is a good indication that in fact our implementation satisfies $\varepsilon = 1$. This same technique also applies to, e.g., the Gaussian mechanism (Fig. 12).