Private Protocols for U-Statistics in the Local Model and Beyond

James Bell, Aurélien Bellet, Adrià Gascón, Tejas Kulkarni

Private Protocols for U-Statistics in the Local Model and Beyond

James Bell* Aurélien Bellet† Adrià Gascón ‡ Tejas Kulkarni §

Abstract

In this paper, we study the problem of computing U-statistics of degree 2, i.e., quantities that come in the form of averages over pairs of data points, in the local model of differential privacy (LDP). The class of U-statistics covers many statistical estimates of interest, including Gini mean difference, Kendall's tau coefficient and Area under the ROC Curve (AUC), as well as empirical risk measures for machine learning problems such as ranking, clustering and metric learning. We first introduce an LDP protocol based on quantizing the data into bins and applying randomized response, which guarantees an ϵ-LDP estimate with a Mean Squared Error (MSE) of $O(1/\sqrt{n}\epsilon)$ under regularity assumptions on the U-statistic or the data distribution. We then propose a specialized protocol for AUC based on a novel use of hierarchical histograms that achieves MSE of $O(\alpha^3/n\epsilon^2)$ for arbitrary data distribution. We also show that 2-party secure computation allows to design a protocol with MSE of $O(1/n\epsilon^2)$, without any assumption on the kernel function or data distribution and with total communication linear in the number of users n. Finally, we evaluate the performance of our protocols through experiments on synthetic and real datasets.

1 Introduction

The problem of collecting aggregate statistics from a set of n users in a way that individual contributions remain private even from the data analysts has recently attracted a lot of interest. In the popular *local model* of differential privacy (LDP) (Duchi et al., 2013; Kairouz et al., 2014), users apply a local randomizer to their private input before sending it to an

*The Alan Turing Institute. jbell@turing.ac.uk. Work supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1, and the UK Government's Defence & Security Programme in support of the Alan Turing Institute.

†INRIA. aurelien.bellet@inria.fr. Work supported by grants ANR-16-CE23-0016-01 and ANR-18-CE23-0018-03, by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 825081 COMPRISE and by a grant from CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020. A. B. thanks Jan Ramon for useful discussions.

‡Google. adriagascon@gmail.com. Work part done when A. G. was at The Alan Turing Institute and Warwick University, and supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1, and the UK Government's Defence & Security Programme in support of the Alan Turing Institute.

§Aalto University, Helsinki, Finland. T. K. thanks Graham Cormode for arranging a trip to Inria, Lille. His visit was supported by Marie Curie Grant 618202.

untrusted aggregator. In this context, most work has focused on computing quantities that are separable across individual users, such as sums and histograms (see Bassily and Smith, 2015; Wang et al., 2017; Kulkarni et al., 2019; Cormode et al., 2018; Bassily et al., 2017, and references therein).

In this paper, we study the problem of privately computing U-statistics of degree 2, which generalize sample mean statistics to *averages over pairs of data points*. Let $x_1, \ldots, x_n$ be a set of n data points drawn i.i.d. from an unknown ditribution μ over a (discrete or continuous) domain $\mathcal{X}$. The U-statistic of degree 2 with kernel f, given by $U_{f,n} = \frac{2}{n(n-1)} \sum_{i<j} f(x_i, x_j)$, is an unbiased estimate of $U_f = \mathbb{E}_{x,x' \sim \mu}[f(x, x')]$ with minimum variance (Hoeffding, 1948). The class of U-statistics covers many statistical estimates of interest, including sample variance, Gini mean difference, Kendall's tau coefficient, Wilcoxon Mann-Whitney hypothesis test and Area under the ROC Curve (AUC) (Lee, 1990; Mann and Whitney, 1947; Faivishevsky and Goldberger, 2008). They are also commonly used as empirical risk measures for machine learning problems such as ranking, clustering and metric learning (Kar et al., 2013; Clémençon et al., 2016).

Interestingly, private estimation of U-statistics in the LDP model for arbitrary kernel functions f and data distributions μ cannot be straightforwardly addressed by resorting to standard local randomizers such as the Laplace mechanism or randomized response. Indeed, one cannot apply the local randomizer to the terms of the sum based on the sensitivity of f (as each term is shared across two users), and perturbing the inputs themselves can lead to large errors when passed through the (potentially discontinuous) function f.

In this work, we design and analyze several protocols for computing U-statistics with privacy and utility guarantees. More precisely:

1. We introduce a generic LDP protocol based on quantizing the data into k bins and applying k-ary randomized response. We show that under an assumption on either the kernel function f or the data distribution μ, the aggregator can construct an ϵ-LDP estimate of $U_{f,n}$ with a Mean Squared Error (MSE) of $O(1/\sqrt{n}\epsilon)$.

2. For the case of the AUC on a domain of size 2^α, whose kernel does not satisfy the regularity assumption required by our previous protocol, we design a specialized protocol based on hierarchical histograms that achieves MSE $O(\alpha^2 \log(1/\delta)/n\epsilon^2)$ under (ϵ, δ)-LDP and $O(\alpha^3/n\epsilon^2)$ under ϵ-LDP, for arbitrary data distribution.

3. Under a slight relaxation of the local model in which we allow pairs of users i and j to compute a randomized version of $f(x_i, x_j)$ with 2-party secure computation, we show that we can design a protocol with MSE of $O(1/n\epsilon^2)$, without any assumption on the kernel function or data distribution and with constant communication for each of the n users.

4. To evaluate the practical performance of the proposed protocols, we present some experiments on synthetic and real datasets for the task of computing AUC and Kendall's tau coefficient.

The paper is organized as follows. Section 2 gives some background on U-statistics and local differential privacy. In Section 3 we present a generic LDP protocol based on

randomizing quantized inputs. Section 4 introduces a specialized LDP protocol for computing the Area under the ROC Curve (AUC). In Section 5, we introduce a generic protocol which operates in a slightly relaxed version of the LDP model where users can run secure 2-party computation. We present some numerical experiments in Section 6, and discuss some future work in Section 7.

2 Background

In this section, we introduce some background on U-statistics and local differential privacy.

2.1 U-Statistics

2.1.1 Definition and Properties

Let μ be an (unknown) distribution over an input space $\mathcal{X}$ and $f : \mathcal{X}^2 \to \mathbb{R}$ be a pairwise function (assumed to be symmetric for simplicity) referred to as the *kernel*. Given a sample $\mathcal{S} = \{x_i\}_{i=1}^n$ of n observations drawn from μ, we are interested in estimating the following population quantity:

$$U_f = \mathbb{E}_{X_1, X_2 \sim \mu}[f(X_1, X_2)]. \tag{1}$$

Definition 1 (Hoeffding, 1948)**.** *The U-statistic of degree 2 with kernel f is given by*

$$U_{f,n} = \frac{1}{\binom{n}{2}} \sum_{i<j} f(x_i, x_j). \tag{2}$$

$U_{f,n}$ is an unbiased estimate of U_f. Denoting by $\zeta_1 = \mathrm{Var}(f(x_1, X_2) \mid x_1))$ and $\zeta_2 = \mathrm{Var}(f(X_1, X_2)$, its variance is given by (Hoeffding, 1948; Lee, 1990):

$$\mathrm{Var}(U_{f,n}) = \frac{1}{\binom{n}{2}}(2(n-2)\zeta_1 + \zeta_2). \tag{3}$$

The above variance is of $O(1/n)$ and is optimal among all unbiased estimators of U_f that can be computed from $\mathcal{S}$. This incurs a complex dependence structure, as each data point appears in $n-1$ pairs. The statistical behavior of U-statistics can be investigated using linearization techniques (Hoeffding, 1948) and decoupling methods (de la Pena and Giné, 1999), which provide tools to reduce their analysis to that of standard i.i.d. averages. One may refer to (Lee, 1990) for asymptotic theory of U-statistics, to (Van Der Vaart, 2000) (Chapter 12 therein) and (de la Pena and Giné, 1999) for nonasymptotic results, and to (Clémençon et al., 2008, 2016) for an account of U-statistics in the context of machine learning and empirical risk minimization.

2.1.2 Motivating Examples

U-statistics are commonly used as point estimators of various global properties of distributions, as well as in statistical hypothesis testing (Lee, 1990; Mann and Whitney, 1947; Faivishevsky and Goldberger, 2008). They also come up as empirical risk measures in machine learning

problems with pairwise loss functions such as bipartite ranking, metric learning and clustering. Below, we give some concrete examples of U-statistics of broad interest to motivate our private protocols.

Gini mean difference. This is a classic measure of dispersion which is often seen as more informative than the variance for some distributions (Yitzhaki, 2003). Letting $\mathcal{X} \subset \mathbb{R}$, it is defined as

$$G = \frac{1}{\binom{n}{2}} \sum_{i<j} |x_i - x_j|, \tag{4}$$

which is a U-statistic of degree 2 with kernel $f(x_i, x_j) = |x_i - x_j|$. Gini coefficient, the most commonly used measure of inequality, is obtained by multiplying G by $(n-1)/2 \sum_{i=1}^n x_i$.

Remark 1. *The variance of a sample, obtained by replacing the absolute difference by the squared difference in* (4)*, is also a U-statistic. However we note that computing the variance can be achieved by computing two sums of locally computable variables (x_i and x_i^2), which can be done with existing LDP protocols.*

Rényi-2 entropy. Also known as collision entropy, this provides a measure of entropy between Shannon's entropy and min entropy which is used in many applications involving discrete distributions (see Acharya et al., 2015, and references therein). It is given by

$$H_2 = -\ln\left(\frac{1}{\binom{n}{2}} \sum_{i<j} \mathbb{I}[x_i = x_j]\right). \tag{5}$$

The expression inside the log is a U-statistic of degree 2 with kernel $f(x_i, x_j) = \mathbb{I}[x_i = x_j]$.

Kendall's tau coefficient. This statistic measures the ordinal association between two variables and is often used as a test statistic to answer questions such as "does a higher salary make one happier?". In learning to rank applications, it is used to evaluate the extent to which a predicted ranking correlates with the (human-generated) gold standard (see e.g., Joachims, 2002; Lapata, 2006). Formally, assuming continuous variables for simplicity, let $\mathcal{X} \subset \mathbb{R}^2$ and $\mathcal{S} = \{x_i = (y_i, z_i)\}_{i=1}^n$. For any $i < j$, the pairs $x_i = (y_i, z_i)$ and $x_j = (y_j, z_j)$ are said be *concordant* if $(y_i > y_j) \wedge (z_i > z_j)$ or $(y_i < y_j) \wedge (z_i < z_j)$, and *discordant* otherwise. Let C and D be the number of concordant and discordant pairs in $\mathcal{S}$. Kendall rank correlation coefficient is defined as:

$$\tau = \frac{C - D}{C + D} = \frac{1}{\binom{n}{2}} \sum_{i<j} \operatorname{sign}(y_i - y_j) \operatorname{sign}(z_i - z_j), \tag{6}$$

which is a U-statistic of degree 2 with kernel $f(x_i, x_j) = \operatorname{sign}(y_i - y_j) \operatorname{sign}(z_i - z_j)$.[1]

Area under the ROC curve (AUC). In binary classification with class imbalance, the Receiver Operating Characteristic (ROC) gives the true positive rate with respect to the false positive rate of a predictor at each possible decision threshold. The AUC is a popular summary of the ROC curve which gives a single, threshold-independent measure of the classifier goodness which corresponds to the probability that the predictor assigns a higher score to a randomly chosen positive point than to a randomly chosen negative one. AUC

[1] One can easily modify the kernel to account for ties.

is widely used as an evaluation metric in machine learning (Bradley, 1997; Herschtal and Raskutti, 2004). Formally, let $\mathcal{X} \subset \mathbb{R} \times \{-1, 1\}$ and $\mathcal{S} = \{x_i = (s_i, y_i)\}_{i=1}^n$ where for each data point i, $s_i \in \mathbb{R}$ is the score assigned to point i and $y_i \in \{-1, 1\}$ is its label. For convenience, let $\mathcal{S}^+ = \{s_i : y_i = 1\}$ and $\mathcal{S}^- = \{s_i : y_i = -1\}$ and let $n^+ = |\mathcal{S}^+|$ and $n^- = |\mathcal{S}^-|$. The AUC is given by

$$AUC = \frac{1}{n^+ n^-} \sum_{s_i \in \mathcal{S}^+} \sum_{s_j \in \mathcal{S}^-} \mathbb{I}[s_i > s_j], \tag{7}$$

where $\mathbb{I}[\sigma]$ is an indicator variable outputting 1 if the predicate σ is true and 0 otherwise. Up to a $\binom{n}{2}/n^+n^-$ factor, it is easy to see that AUC is a U-statistic of degree 2 with kernel $f(x_i, x_j) = \mathbb{I}[s_i > s_j \wedge y_i > y_j] + \mathbb{I}[s_i < s_j \wedge y_i < y_j]$.

Machine learning with pairwise losses. Many machine learning problems involve loss functions that operate on pairs of points (Kar et al., 2013; Clémençon et al., 2016). This is the case for instance in metric learning (Bellet et al., 2015), bipartite ranking (Clémençon et al., 2008) and clustering (Clémençon, 2014). Empirical risk minimization problems have therefore the following generic form:

$$\min_{\theta \in \Theta} \frac{1}{\binom{n}{2}} \sum_{i<j} \ell_\theta(x_i, x_j), \tag{8}$$

where $\theta \in \Theta$ are model parameters. The objective function in (8), as well as its gradient, are U-statistics of degree 2 with kernels ℓ_θ and $\nabla_\theta \ell_\theta$ respectively.

2.2 Local Differential Privacy

The classic *centralized* model of differential privacy assumed the presence of a trusted aggregator which processes the private information of individuals and releases a perturbed version of the result. The *local* model instead captures the setting where individuals do not trust the aggregator and randomize their input locally before sharing it. This model has received wide industrial adoption (Erlingsson et al., 2014; Fanti et al., 2016; Differential Privacy Team, Apple, 2017; Ding et al., 2017).

Definition 2 (Duchi et al., 2013)**.** *A local randomizer $\mathcal{R}$ is (ϵ, δ)-locally differentially private (LDP) if for all pairs $x, x' \in \mathcal{X}$ and all possible output O in the range of $\mathcal{R}$, we have*

$$Pr[\mathcal{R}(x) = O] \leq e^\epsilon Pr[\mathcal{R}(x') = O] + \delta.$$

The special case $\delta = 0$ is called pure ϵ-LDP.

Most work in LDP aims to compute quantities that are separable across individual inputs, such as sums and histograms (see Bassily and Smith, 2015; Wang et al., 2017; Kulkarni et al., 2019; Cormode et al., 2018; Bassily et al., 2017, and references therein). In contrast, our goal is to design LDP protocols for computing U-statistics, where each term involves a pair of inputs.

Algorithm 1: LDP algorithm based on quantization and private histograms

Public Parameters: Privacy budget ϵ, quantization scheme π, number of bins k.
Input: $(x_i \in \mathcal{X})_{i\in[n]}$
Output: Estimate $\widehat{U}_{f,n}$ of U_f

for *each user* $i \in [n]$ **do**
  Form quantized input $\pi(x_i) \in [k]$
  For $\beta = k/(k + e^\epsilon - 1)$, generate $\tilde{x}_i \in [k]$ such that

$$P(\tilde{x}_i = i) = \begin{cases} 1-\beta & \text{for } i = \pi(x_i), \\ \beta/k & \text{for } i \neq \pi(x_i), \end{cases} \tag{9}$$

  Send $\tilde{x}_i$ to the aggregator
end
Return $\widehat{U}_{f,n}$ computed from $\tilde{x}_1, \ldots, \tilde{x}_n$ and β

3 Generic LDP Protocol from Quantization

Discrete inputs. We first consider the case of discrete inputs taking one of k values. The possible values of the kernel function can be written as a matrix $A \in \mathbb{R}^{k\times k}$ where $A_{ij} = f(i, j)$. In this case, we can set the local randomizer $\mathcal{R}$ to be k-ary randomized response to generate a perturbed version $\mathcal{R}(x_i)$ of each input x_i. Let e_i denote the vector of length k with a one in the i-th position and 0 elsewhere. For each perturbed input in one-hot encoding form $e_{\mathcal{R}(x_i)}$ we can deduce an unbiased estimate of e_{x_i}. As the discrete U-statistic is a linear function of each of these vectors, computing it on these unbiased estimates gives an unbiased estimate $\widehat{U}_{f,n}$ which can be written as:

$$\widehat{U}_{f,n} = \frac{1}{\binom{n}{2}} \sum_{1\le i<j\le n} \widehat{f}_A(\mathcal{R}(x_i), \mathcal{R}(x_j)),$$

and is itself a U-statistic with kernel $\widehat{f}_A$ given by

$$\widehat{f}_A(\mathcal{R}(x_1), \mathcal{R}(x_2)) = (1-\beta)^{-2}(e_{\mathcal{R}(x_1)} - b)^T A(e_{\mathcal{R}(x_2)} - b),$$

where $1-\beta$ is the probability of returning the true input in k-ary randomized response (see Eq. 9) and b is the vector of length k with every entry β/k. Details and analysis of this process, leveraging results on the variance of U-statistics (Hoeffding, 1948; Lee, 1990), can be found in Section A.1 of the supplementary material. The resulting bounds on the variance of $\widehat{U}_{f,n}$ are summarized in the following theorem.

Theorem 1. *If $f(x, x') \in [0, 1]$ for all x, x', then*

$$\mathrm{Var}(\widehat{U}_{f,n}) \le \frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4}.$$

In order to achieve ϵ-LDP with a fixed k this becomes,

$$\mathrm{Var}(\widehat{U}_{f,n}) \approx \frac{(1+k/\epsilon)^2}{n} + \frac{(1+k/\epsilon)^4}{2n^2} \approx \frac{k^2}{n\epsilon^2},$$

Continuous inputs. For U-statistics on discrete domains, e.g. Renyi-2 entropy, the above strategy can be applied directly. Possibly more importantly however, this then leads to a natural protocol for the continuous case. In this protocol (see Algorithm 1), the local randomizer proceeds by quantizing the input into k bins (for instance using simple or randomized rounding) before applying the previous procedure.

There are two sources of error in this protocol. The first one is due to the randomization needed to satisfy LDP in the quantized domain as bounded in Theorem 1. The second source of error is due to quantization. In order to control this error in a nontrivial way, we rely on an assumption on the kernel function (namely, that it is Lipschitz) or the data distribution (namely, that it has Lipschitz density). Under these assumptions and using an appropriate variant of the kernel function on the quantized domain, we show that we can bound the error with respect to the original domain by a term in $O(1/k^2)$ (see Section A.2 of the supplementary material). This leads to the following result.

Theorem 2. *For simplicity, assume bounded domain $\mathcal{X} = [0, 1]$ and kernel values $f(x, y) \in [0, 1]$ for all $x, y \in \mathcal{X}$. Let π correspond to simple rounding, $\epsilon > 0$, $k \geq 1$ and $\beta = k/(k+e^{\epsilon}-1)$. Then Algorithm 1 satisfies ϵ-LDP. Furthermore:*

- *If f is L_f-Lipschitz in each of its arguments, then $\texttt{MSE}(\widehat{U}_{f,n})$ is less than or equal to*

$$\frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4} + \frac{L_f^2}{2k^2}.$$

- *If $d\mu/d\lambda$ is L_μ-Lipschitz w.r.t. some measure λ, then $\texttt{MSE}(\widehat{U}_{f,n})$ is less than or equal to*

$$\frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4} + \frac{4L_\mu^2}{k^2} + \frac{4L_\mu^4}{k^4}.$$

Setting k so as to balance the quantization and estimation errors leads to the following corollary.

Corollary 1. *Under the conditions of Theorem 2, for $\epsilon \leq 1$ and large enough n, taking $k = n^{1/4}\sqrt{L\epsilon}$ leads to $\texttt{MSE}(\widehat{U}_{f,n}) = O(L/\sqrt{n}\epsilon)$, where L corresponds to L_f or L_μ depending on the assumption.*

This result gives concrete error bounds for U-statistics whose kernel is Lipschitz, for arbitrary data distributions. One important example is the Gini mean difference, whose corresponding kernel $f(x_i, x_j) = |x_i - x_j|$ is 1-Lipschitz. On the other hand, for U-statistics with non-Lipschitz kernels, the data distribution must be sufficiently smooth (if not, it is easy to construct cases that make the algorithm fail).

4 Locally Private AUC

In this section, we describe an algorithm for computing AUC (7), whose kernel is discontinuous and therefore non-Lipschitz. We assume $\mathcal{X}$ to be an ordered domain of size d, that is with each datum in $[0..d-1]$. Note that all data is in practice discrete when represented in finite

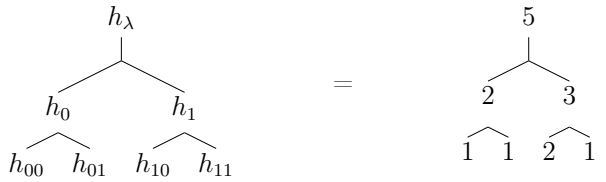

Figure 1: Hierarchical histogram h for multiset $\{0, 1, 2, 2, 3\}$ over the domain $\{0, 1, 2, 3\}$.

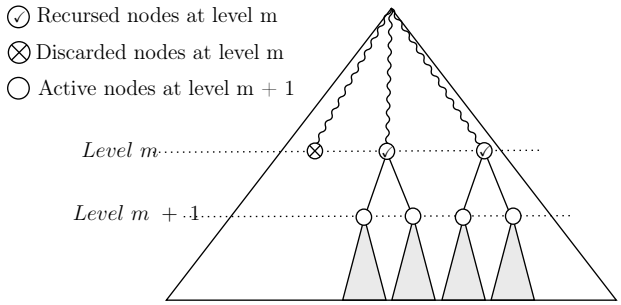

Figure 2: Our algorithm can be seen as a breath-first traversal of a tree, where at each level some nodes are selected for their subtrees to be explored further.

precision, so this is general. For simplicity of presentation we will assume that (i) $d = 2^\alpha$ for some integer α, and (ii) that the classes of the data, the y_i, are public.

Our solution for computing AUC in the local model relies on a hierarchical histogram construction that has been considered in previous works for private collection of high-dimensional data (Chan et al., 2012), heavy hitters (Bassily et al., 2017), and range queries (Kulkarni et al., 2019). A hierarchical histogram is essentially a tree data structure on top of a histogram where each internal node is labelled with the sum of the values in the interval covered by it (see Figure 1). That allows to answer any range query about u by checking the value associated with $O(\log |u|)$ nodes in the tree. We first define an exact version of such hierarchical histograms and explain how to compute AUC from one.

Notation on trees. We represent a binary tree h of depth α with integer node labels as a total mapping from a prefix-closed set of binary strings of length at most α to the integers. We refer to the i-th node in level l of the tree by the binary representation of i padded to length l from the left with zeros. With this notation, h_λ is the label of the root node, as we use λ to denote the empty string, h_0 (resp. h_1) is the integer label of the left (resp. right) child of the root of h, and in general h_p is the label of the node at path p from the root, i.e. the label of the node reached by following left or right children from the root according to the value of p (0 indicates left and 1 indicates right). Let b_i be the i-th node in the bottom level. For two binary strings $p, p' \in \{0, 1\}^*$ we denote the prefix relation by $p' \preceq p$, and their concatenation as $p \cdot p'$.

Definition 3. *Let $S = \{s_1, \ldots, s_n\}$ be a multiset, with $s_i \in [0..d-1]$. A hierarchical histogram of S is a total mapping $h : \{0, 1\}^{\leq \log(d)} \to \mathbb{Z}$ defined as $h(b) = |\{s \in S \mid \exists b' \in \{0, 1\}^* : b \cdot b' = b_s\}|$. For simplicity, we denote $h(b)$ by h_b.*

Algorithm. We use hierarchical histograms to compute AUC as follows. Let S^+ and S^- be the samples of the positive and negative classes from which we want to estimate AUC. Let h^+ and h^- be hierarchical histograms for S^+ and S^-. Note that $h^+_\lambda = n^+$ and $h^-_\lambda = n^-$. We can now define the unnormalized AUC, denoted UAUC, over hierarchical histograms recursively by letting $\texttt{UAUC}(h^+, h^-, p)$ be 0, if p is a leaf, and otherwise setting:

$$\texttt{UAUC}(h^+, h^-, p) = h^+_{p\cdot 1} h^-_{p\cdot 0} + \sum_{i\in\{0,1\}} \texttt{UAUC}(h^+, h^-, p\cdot i) \ .$$

Thus we have $\texttt{AUC}(S^+, S^-) = \texttt{AUC}(h^+, h^-, \lambda) = \frac{1}{n^+ n^-}\texttt{UAUC}(h^+, h^-, \lambda)$.

The above definition naturally leads to an algorithm that proceeds by traversing the trees h^+, h^- top-down from the root λ, accumulating the products of counts from h^+, h^- at nodes that correspond to entries in h^+ that are bigger than entries in h^-. We now define a differentially private analogue. Later we will describe an efficient frequency oracle which can be used to compute an LDP estimate $\hat{h}$ of a hierarchical histogram h of n values in a domain of size 2^α. This will provide the following necessary properties (i) $\hat{h}$ is *unbiased*, (ii) $\mathrm{Var}(\hat{h}) \le v$, with v defined as $Cn\alpha$ for some small constant C (iii) the $\hat{h}_p$ are pairwise independent and (iv) Each level of $\hat{h}$ is independent of the other levels. Our private algorithm for computing an estimate of UAUC is then defined in terms of parameters n^+ and n^-, v^+ and v^- (bounding the variance of $\hat{h}^+$ and $\hat{h}^-$ respectively), and $a > 1$ is a small number depending on n^+, n^-, α and C.

For a symbol $\aleph$ we write $\aleph^\pm$ to simultaneously refer to $\aleph^+$ and $\aleph^-$. Let $\tilde{h}^\pm_p = \max(\hat{h}^\pm_p, \sqrt{av^\pm}/2)$, i.e. $\tilde{h}^+_p = \max(\hat{h}^+_p, \sqrt{av_+}/2)$ and $\tilde{h}^-_p = \max(\hat{h}^-_p, \sqrt{av_-}/2)$, and let $\tau = a\sqrt{v^- v^+}$. Our private estimate is defined as follows. If p is a leaf then $\widehat{\texttt{UAUC}}(\hat{h}^+, \hat{h}^-, p)$ is 0, else if $\tilde{h}^+_p \tilde{h}^-_p < \tau$ then it is given by

$$\frac{1}{2} \sum_{i\in\{0,1\}} \hat{h}^+_{p\cdot i} \sum_{i\in\{0,1\}} \hat{h}^-_{p\cdot i} \ .$$

Otherwise, it is given recursively by

$$\hat{h}^+_{p\cdot 1}\hat{h}^-_{p\cdot 0} + \sum_{i\in\{0,1\}} \widehat{\texttt{UAUC}}(\hat{h}^+, \hat{h}^-, p\cdot i) \ . \tag{10}$$

As before, this definition leads to an algorithm. Note that the only difference with its non-private analogue is that this procedure does not recurse into subtrees whose contribution to the UAUC is upper bounded sufficiently tightly. More concretely, the server starts by querying $\hat{h}^+, \hat{h}^-$ at the root, namely with $p = \lambda$. If p is a leaf then we return 0 as the AUC. Otherwise, the algorithm checks whether $\tilde{h}^+_p \tilde{h}^-_p < \tau$. If so, then the algorithm concludes that there is not much to gain in exploring the subtrees rooted at $p\cdot 0$ and $p\cdot 1$, and returns $\frac{1}{2}\sum_{i\in 0,1} \hat{h}^+_{p\cdot i} \sum_{i\in 0,1} \hat{h}^-_{p\cdot i}$ as an estimate of $\frac{1}{2}h^+_p h^-_p$. This estimate might seem equivalent to $\frac{1}{2}\hat{h}^+_p \hat{h}^-_p$, but takes the previous form for a technical reason that is made clear in the proof. In this case we call p a *discarded* node. On the other hand, if $\tilde{h}^+_p \tilde{h}^-_p \ge \tau$, the algorithm proceeds as its non-private analogue, accumulating the contribution to the UAUC from the direct subtrees of p and recursing into nodes $p\cdot 0$ and $p\cdot 1$. In this case we refer to p as a *recursed* node. Thus every node $p \in \{0,1\}^{\le\alpha}$ will be either recursed, a leaf or there will be a discarded node p' such that $p' \preceq p$. This is depicted in Figure 2.

Analysis. Note that our algorithm has two sources of error: (i) the one incurred by discarding nodes and (ii) the error in estimating the contribution to the UAUC of the recursed nodes. The threshold τ is carefully chosen to balance these two errors.

Let R^m be the set of nodes recursed on at level m. Our accuracy proof starts by bounding the expected value of $|R^m|$ (see Lemma 4 in Section B.1 of the supplementary) by a quantity B that is independent of m. We now describe a central argument to our accuracy proof, stated in the next theorem. Let E_m^R be the contribution to the error by nodes in R^m. Then, the total contribution to the error by recursed nodes is $E^R = \sum_{m\in[\alpha]} E_m^R$. A useful identity is $\mathbb{E}(E^{R2}) = \sum_{m\in[\alpha]} \mathbb{E}(E_m^{R2})$, as we can bound $\mathbb{E}(E_m^{R2})$, for any m, in terms of B (see detailed proof in the supplementary). Note that this identity follows from $\mathbb{E}(E_m^R E_{m'}^R) = 0$, with $m' > m$. The latter would hold if errors E_m^R and $E_{m'}^R$ were independent, since our frequency oracle is unbiased. However, errors at a given level are not independent of previous levels. However $\mathbb{E}(E_m^R E_{m'}^R) = 0$ because the conditional expectation of $E_{m'}^R$ with respect to the answers of the frequency oracles up to level m' is 0 i.e. $E_1^R, \ldots, E_{m'}^R$ is a martingale difference sequence. The idea of conditioning on previous levels is used several times in our proofs, also to bound the error due to discarded nodes.

Next, we state our accuracy result, which is proven in detail in Section B.1 of the supplementary. Our proof tracks constants: this is important for practical purposes, and we show empirically in Section B.3 that our bound is in fact quite tight.

Theorem 3. *If $\alpha \le \sqrt{n}$ and the following conditions hold:*

1. $\mathbb{E}(\hat{h}_p^{\pm} - h_p^{\pm}) = 0$ *i.e. frequency estimates are unbiased.*
2. $\mathbb{E}((\hat{h}_p^{\pm} - h_p^{\pm})^2) \le v^{\pm}$ *i.e.* `MSE` *of frequency estimator is bounded by $v^{\pm} = Cn^{\pm}\alpha$.*
3. *For distinct $p, p' \in \{0,1\}^{\le\alpha}$ with $|p| = |p'|$, $\hat{h}_p^{\pm}$ and $\hat{h}_{p'}^{\pm}$ are independent i.e. the frequency estimates are pairwise independent.*
4. *For all $m \le \log(d)$, the lists $(\hat{h}_p^{\pm})_{p\in\{0,1\}^{\le m}}$ and $(\hat{h}_p^{\pm})_{p\in\{0,1\}^{>m}}$ are independent of each other.*

Then, `MSE`$(\widehat{\texttt{UAUC}})$ *is given by*

$$Cn^{-}n^{+}\alpha^2\left(2n + (4a+1)\min(n^{-}, n^{+}) + \frac{21\sqrt{2nC\alpha}}{\sqrt{a}-1}\right)$$

Instantiating $\hat{h}$. So far, Theorem 3 does not yield a complete algorithm as it does not specify an algorithm for computing estimates $\hat{h}$ of a hierarchical histogram that satisfy the conditions of Theorem 3. In Section B.2 of the supplementary, we show how to instantiate such an algorithm in a communication-efficient manner by combining ideas from Bassily et al. (2017), in particular the use of the Hadamard transform, with an modified version of the protocol from Kulkarni et al. (2019). This leads to the following result.

Theorem 4. *There is a one-round non-interactive protocol for computing AUC in the local model with* `MSE` *bounded by $O(\alpha^2 \log(1/\delta)/n\epsilon^2)$ under (ϵ, δ)-LDP and $O(\alpha^3/n\epsilon^2)$ under ϵ-LDP. Every user submits one bit, and the server does $O(nlog(d))$ computation and requires $O(log(d))$ additional reconstruction space.*

5 Generic Protocols from 2PC

So far, we have proposed a specialized LDP protocol for AUC, and a generic LDP protocol which requires some assumption on the kernel function or the data distribution to guarantee nontrivial error bounds. We conjecture that no LDP protocol can guarantee nontrivial error for arbitrary kernels and distributions, but we leave this as an open question for future work.

In this section, we slightly relax the model of LDP by allowing pairs of users i and j to compute a randomized version $\tilde{f}(x_i, x_j)$ of their kernel value $f(x_i, x_j)$ with 2-party secure computation (2PC). This gives rise to a computational differential privacy (CDP) model Mironov et al. (2009). Unsurprisingly, we show that in this model we can match the MSE of $O(\frac{\ln(1/\delta)}{n\epsilon^2})$ for computing regular (univariate) averages in the (ϵ, δ)-LDP model by using advanced composition results (Dwork et al., 2010). However, such a protocol requires $O(n^2)$ communication as all pairs of users need to compute $\tilde{f}(x_i, x_j)$ via 2PC, and does not satisfy pure ϵ-DP.

Proposed protocol. To address these limitations, we propose a protocol in which the aggregator asks only a (random) subset of pairs of users (i, j) to submit their randomized kernel value $\tilde{f}(x_i, x_j)$. The idea is to trade-off between the error due to privacy (which increases as more pairs are used, due to budget splitting) and the *subsampling error* (for not averaging over all available pairs). Given a positive integer P (which should be thought of as a small constant independent of n) and assuming n to be even for simplicity, we propose the following protocol:

1. *Subsampling:* The aggregator samples P independent permutations $\sigma_1, \ldots, \sigma_P \in \mathfrak{S}_n$ of the set of users $\{1, \ldots, n\}$. This defines a set of $Pn/2$ pairs $\mathcal{P} = (\sigma_p(2i - 1, 2i))_{p\in[P], 1\le i\le N/2}$.

2. *Perturbation:* For each pair of users $(i, j) \in \mathcal{P}$, users compute $\tilde{f}(x_i, x_j)$ via 2PC and sends it to the aggregator.

3. *Aggregation:* The aggregator computes an estimate of U_f as a function of $\{\tilde{f}(x_i, x_j)\}_{(i,j)\in\mathcal{P}}$.

Analysis. We have the following result for the Laplace mechanism applied to real-valued kernel functions (the extension to randomized response for discrete-valued kernels is straightforward). The proof relies on an exact characterization of the subsampling error by leveraging results on the variance of incomplete U-statistics (Blom, 1976), see Section C.1 of the supplementary for details.

Theorem 5 (2PC subsampling protocol with Laplace mechanism)**.** *Let $\epsilon > 0$, $P \ge 1$ and assume that the kernel f has values in $[0, 1]$. Consider our subsampling protocol above with $\tilde{f}(x_i, x_j) = f(x_i, x_j) + \eta_{ij}$ where $\eta_{ij} \sim Lap(P/\epsilon)$, and the estimate computed as $\widehat{U}_{f,n} = \frac{2}{Pn}\sum_{(i,j)\in\mathcal{P}} \tilde{f}(x_i, x_j)$. Then the protocol satisfies ϵ-CDP, has communication cost of $O(Pn)$ and*

$$\mathtt{MSE}(\widehat{U}_{f,n}) = \frac{2}{Pn}\Big(2(P-1)\big(1 - \frac{1}{n-1}\big)\zeta_1 + \big(1 + \frac{P-1}{n-1}\big)\zeta_2\Big) + \frac{2P}{n\epsilon^2},$$

where ζ_1 and ζ_2 are defined as in (3).

The MSE in Theorem 5 is of $O(\frac{1}{Pn} + \frac{P}{n\epsilon^2})$. Remarkably, this shows that the $O(1/n)$ variance of the estimate that uses all pairs is preserved when subsampling only $O(n)$ pairs. This is made possible by the strong dependence between the $O(n^2)$ terms of the original U-statistic. As expected, P rules a trade-off between the error due to subsampling and the error due to privacy: the larger P, the smaller the former but the larger the latter (as each user must split its budget across P pairs). The optimal value of P depends on the kernel function and the data distribution (through ζ_1 and ζ_2) on the one hand, and the privacy budget ϵ on the other hand. This trade-off, along with the optimality of the proposed subsampling schemes, are discussed in more details Section C.1 of the supplementary material. In practice and as illustrated in our experiments, P can be set to a small constant.

Implementing 2PC. Securely computing the randomized kernel value $\tilde{f}(x_i, x_j)$ can be done efficiently for many kernel functions and local randomizers of interest, as the number of parties involved is limited to 2. We assume semi-honest parties (see Goldreich, 2004, for a definition of this threat model). A suitable 2PC technique in this application are garbled circuits (Yao, 1986; Lindell and Pinkas, 2009; Evans et al., 2018), which are well-suited to compute Boolean comparisons as required in several of the kernels mentioned in Section 2.1.2. The circuits for computing the kernels can then be extended with output perturbation following ideas from Dwork et al. (2006) and Champion et al. (2019). We refer to Section C.2 of the supplement for details on design and complexity.

Remark 2 (Beyond 2PC)**.** *One could further relax the model to allow multi-party secure computation with more than two parties, e.g. by extending the garbled circuit computing the kernel with secure aggregation over the Pn pairs before performing output perturbation. This would recover the utility of centralized DP at the cost of much more computation and quadratic communication, which is not practical, as well as robustness. More interesting trade-offs may be achieved by securely aggregating small subsets of pairs. We leave the careful analysis of such extensions to future work.*

6 Experiments

AUC protocol. We use the Diabetes dataset (Strack et al., 2014) for the binary classification task of determining whether a patient will be readmitted in the next 30 days after being discharged. We train a logistic regression model s which is used to score data points in $[0, 1]$, and apply our protocol to privately compute AUC on the test set. Patients readmitted before 30 days form the positive class. Class sizes are shown in Figure 3. Class information is not considered sensitive, as opposed to the score $s(x)$ on private user data $s(x)$, which includes detailed medical information. Figure 3 shows the standard error achieved by our protocol for different values of the domain size d. For each value of d we run our protocol with inputs $s(\texttt{fp}(s(x_i), d))$, where `fp` denotes a discretization into the domain $[0..d-1]$. The plot shows that $d = 2^9$ gives the smallest error, and that increasing ϵ beyond 5 does not improve results significantly. Recall that the error of our AUC protocol depends on the size of the smallest class, which is quite small here (only 661 examples).

Protocol from 2PC. We use the Tripadvisor dataset (Wang, 2010). The dataset consists of user ratings (from scale -1 to 5) for hotels in San Francisco over many service quality metrics such as room service, location, room cleanliness, front desk service etc. After discarding

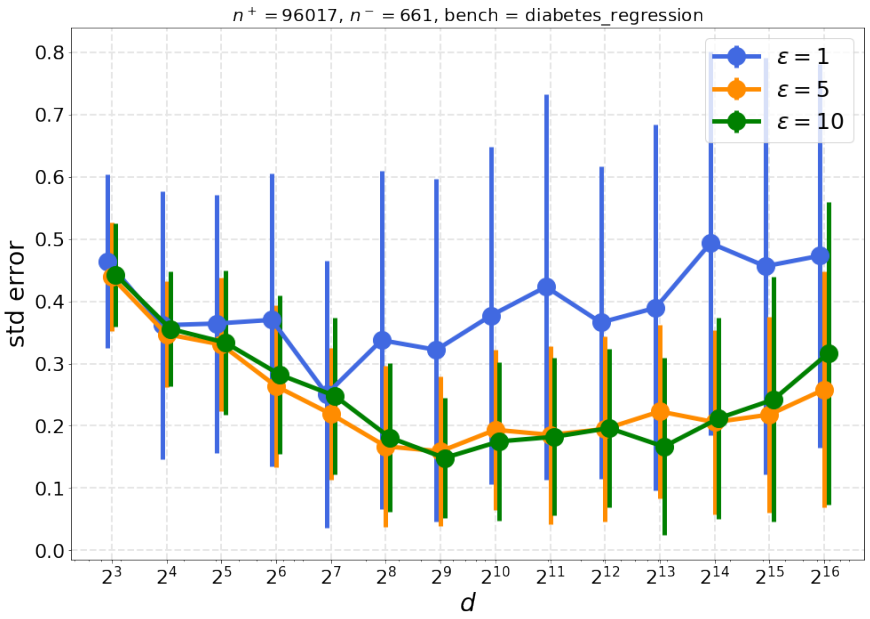

Figure 3: Mean and standard deviation (over 20 runs) of the absolute error of our AUC protocol on the scores of a logistic regression model trained on a Diabetes dataset.

the records with missing values, we have over 246K records. Let $x_i = (y_i, z_i)$ be ratings given by user i to the room (y_i) and the cleanliness (z_i). Our goal is to privately estimate the Kendall's tau coefficient (KTC), see (6). The true value of the KTC between the two variables of interest is 0.8. For this experiment, the 2PC primitive to compute $\tilde{f}(x_i, x_j)$ is simulated. The main purpose here is to empirically evaluate the privacy-utility trade-off of our protocol based on subsampling, compared to the protocol that computes all pairs and relies on advanced composition (we set $\delta = 1e^{-8}$ for the latter). The results shown in Figure 4 show that our subsampling protocol achieves lower error by roughly an order of magnitude, despite providing a stronger pure LDP guarantee and requiring only linear communication in n. As predicted by our analysis, $P = 1$ is best in high privacy regimes (small ϵ) where the error due to privacy dominates the error due to subsampling. We also see that $P > 1$ can be used to reduce the overall error in low privacy regimes.

7 Concluding Remarks

In this paper, we introduced several protocols for computing U-statistics from private user data. Our methods cover many statistical quantities of broad interest which were not addressed by previous private protocols. While we focused on pairwise U-statistics for ease of exposition, our ideas can be extended to U-statistics of higher degrees such as the Volume Under the ROC Surface (VUS), the generalization of AUC to multipartite ranking (Clémençon et al., 2013).

Our protocol in Section 3 is not tight in many situations. In the case of sum, and possibly in the case of the Gini coefficient, it would be more accurate if randomized rounding were used instead of simple rounding. We leave this investigation for later work.

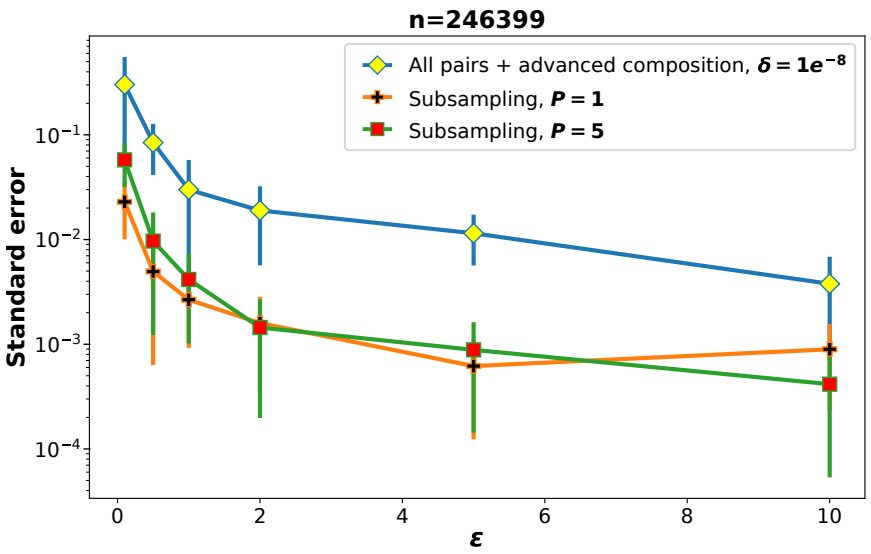

Figure 4: Mean and standard deviation (over 20 runs) of the absolute error in KTC on Tripadvisor dataset.

A promising direction for future work is to develop private multi-party algorithms for learning with pairwise losses (Kar et al., 2013; Clémençon et al., 2016) by combining private stochastic gradient descent for standard empirical risk minimization (Bassily et al., 2014; Shokri and Shmatikov, 2015) and our protocols to compute the gradient estimates.

References

Acharya, J., Orlitsky, A., Suresh, A. T., and Tyagi, H. (2015). The Complexity of Estimating Rényi Entropy. In *SODA*.

Bassily, R., Nissim, K., Stemmer, U., and Thakurta, A. G. (2017). Practical locally private heavy hitters. In *NIPS*.

Bassily, R. and Smith, A. (2015). Local, private, efficient protocols for succinct histograms. In *STOC*.

Bassily, R., Smith, A. D., and Thakurta, A. (2014). Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. In *FOCS*.

Bellet, A., Habrard, A., and Sebban, M. (2015). *Metric Learning*. Morgan & Claypool Publishers.

Blom, G. (1976). Some properties of incomplete U-statistics. *Biometrika*, 63(3):573–580.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

Champion, J., Shelat, A., and Ullman, J. (2019). Securely sampling biased coins with applications to differential privacy. *IACR Cryptology ePrint Archive*, 2019:823.

Chan, T. H., Shi, E., and Song, D. (2012). Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography*.

Clémençon, S. (2014). A statistical view of clustering performance through the theory of U-processes. *Journal of Multivariate Analysis*, 124:42–56.

Clémençon, S., Bellet, A., and Colin, I. (2016). Scaling-up Empirical Risk Minimization: Optimization of Incomplete U-statistics. *Journal of Machine Learning Research*, 13:165–202.

Clémençon, S., Lugosi, G., and Vayatis, N. (2008). Ranking and empirical risk minimization of U-statistics. *The Annals of Statistics*, 36(2):844–874.

Clémençon, S., Robbiano, S., and Vayatis, N. (2013). Ranking data with ordinal labels: Optimality and pairwise aggregation. *Machine Learning*, 91(1):67–104.

Cormode, G., Kulkarni, T., and Srivastava, D. (2018). Marginal release under local differential privacy. In *SIGMOD*.

Damgård, I., Pastro, V., Smart, N. P., and Zakarias, S. (2011). Multiparty computation from somewhat homomorphic encryption. *IACR Cryptology ePrint Archive*, 2011:535.

de la Pena, V. and Giné, E. (1999). *Decoupling: from Dependence to Independence*. Springer.

Differential Privacy Team, Apple (2017). Learning with privacy at scale.

Ding, B., Kulkarni, J., and Yekhanin, S. (2017). Collecting telemetry data privately. In *NIPS*.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. (2013). Local privacy and statistical minimax rates. In *FOCS*.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006). Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer.

Dwork, C., Rothblum, G. N., and Vadhan, S. (2010). Boosting and Differential Privacy. In *FOCS*.

Erlingsson, U., Pihur, V., and Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *CCS*.

Evans, D., Kolesnikov, V., and Rosulek, M. (2018). A pragmatic introduction to secure multi-party computation. *Foundations and Trends in Privacy and Security*, 2(2-3):70–246.

Faivishevsky, L. and Goldberger, J. (2008). ICA based on a Smooth Estimation of the Differential Entropy. In *NIPS*.

Fanti, G., Pihur, V., and Erlingsson, Ú. (2016). Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. In *PoPETs*.

Goldreich, O. (2004). *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press.

Goldreich, O., Micali, S., and Wigderson, A. (1987). How to play any mental game or A completeness theorem for protocols with honest majority. In *STOC*, pages 218–229. ACM.

Herschtal, A. and Raskutti, B. (2004). Optimising area under the ROC curve using gradient descent. In *ICML*.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *Annals of Mathematics and Statistics*, 19:293–325.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *KDD*.

Kairouz, P., Oh, S., and Viswanath, P. (2014). Extremal mechanisms for local differential privacy. In *NIPS*.

Kar, P., Sriperumbudur, B. K., Jain, P., and Karnick, H. (2013). On the Generalization Ability of Online Learning Algorithms for Pairwise Loss Functions. In *ICML*.

Kulkarni, T., Cormode, G., and Srivastava, D. (2019). Answering range queries under local differential privacy. In *SIGMOD*.

Lapata, M. (2006). Automatic Evaluation of Information Ordering: Kendall's Tau. *Computational Linguistics*, 32(4):471–484.

Lee, A. (1990). *U-statistics: Theory and practice*. Marcel Dekker, Inc., New York.

Lindell, Y. and Pinkas, B. (2009). A proof of security of yao's protocol for two-party computation. *J. Cryptology*, 22(2):161–188.

Mann, H. B. and Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics*, 18(1):50–60.

Mironov, I., Pandey, O., Reingold, O., and Vadhan, S. P. (2009). Computational Differential Privacy. In *CRYPTO*.

Shokri, R. and Shmatikov, V. (2015). Privacy-preserving deep learning. In *CCS*.

Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., and Clore, J. N. (2014). Diabetes data. `https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008`.

Van Der Vaart, A. W. (2000). *Asymptotic Statistics*. Cambridge University Press.

Wang, H. (2010). Trip advisor data. `http://www.preflib.org/data/combinatorial/trip/`.

Wang, T., Blocki, J., Li, N., and Jha, S. (2017). Locally differentially private protocols for frequency estimation. In *USENIX Security Symposium*.

Yao, A. C. (1986). How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167. IEEE Computer Society.

Yitzhaki, S. (2003). Gini's Mean Difference: A Superior Measure of Variability for Non-Normal Distributions. *Metron International Journal of Statistics*, 61(2):285–316.

A Details and Proofs for Generic LDP Protocol

We start by introducing some notations. We denote by k the number of bins of the quantization and by $\pi : \mathcal{X} \to [k]$ the quantization scheme such that for any data point $x \in \mathcal{X}$, $\pi(x)$ denotes the quantized version of x (i.e., its image under π). Let e_i denote the vector of length k with a one in the i-th position and 0 elsewhere. A kernel function f_A on the quantized domain $[k]$ is fully described by a matrix $A \in \mathbb{R}^{k\times k}$ such that $f_A(i,j) = A_{i,j} = e_i^T A e_j$. We denote by $U_{A,\pi} = \mathbb{E}_{\mu\times\mu}[A_{\pi(x),\pi(y)})]$ the quantized analogue to the quantity U_f.

The proposed protocol, described in Algorithm 1, applies generalized randomized response on data quantized with π and uses this to compute an unbiased estimate of a quantized U-statistic. The choice of the quantized kernel A will be discussed below. Crucially, there are two sources of error in this protocol. More precisely, the mean squared error of the estimate $\widehat{U}_{f,n}$ returned by Algorithm 1 can be bounded as follows:

$$\texttt{MSE}(\widehat{U}_{f,n}) \leq (U_f - U_{A,\pi})^2 + \mathbb{E}[(U_{A,\pi} - \widehat{U}_{f,n})^2]. \tag{11}$$

The first term corresponds to the error due to quantization, while the second one is the estimation error due to randomization needed to satisfy local differential privacy. The latter will increase with k, thereby constraining k to remain reasonably small. We will thus need to rely on assumptions on either the kernel or the data distribution to be able to control the error due to quantization.

In line with the error decomposition in (11), we conduct our analysis by considering the effect of sampling and randomization together. Therefore, we will not provide a direct bound on the error between our estimate and the U-statistic of the sample, but directly with respect to the population quantity U_f. We now show how to control the two sources of error, which are easily combined to yield Theorem 2.

A.1 Bounding the Error of Randomized Response on Discrete Domain

In this part, we bound the second term in (11): we consider that the data is discrete ($\mathcal{X} = [k]$) and derive error bounds for the estimate $\widehat{U}_{f,n}$ with respect to $U_{A,\pi}$ for a given kernel function $\tilde{f}(i,j) = A_{i,j}$. We propose to use the generalized randomized response mechanism as our local randomizer $\mathcal{R}$. We introduce some notations. Let β be the probability of $\mathcal{R}$ selecting a response uniformly at random, i.e. let $\mathbb{P}(\mathcal{R}(x) = y) = \beta/k + (1-\beta)\chi_{x=y}$, let b be the vector of length k with every entry β/k. For convenience, we denote the data sample by $x_1, \ldots, x_n \in [k]$. Note that these data points are drawn i.i.d. from a distribution D over $[k]$ (which follows from μ and π) such that $\mathbb{P}(x_i = j) = D_j$.

With these notations, we write the expected value of the discretized kernel computed directly on the randomized data points:

$$\begin{aligned}\mathbb{E}[(e_{\mathcal{R}(x_1)})^T A e_{\mathcal{R}(x_2)}] &= \mathbb{E}(((1-\beta)e_{x_1} + b)^T A((1-\beta)e_{x_2} + b)) \\ &= \mathbb{E}((1-\beta)^2 e_{x_1}^T A e_{x_2} + (1-\beta)(e_{x_1} + e_{x_2})^T Ab + b^T Ab).\end{aligned}$$

This is a biased estimator of $f_A(x_1, x_2) = e_{x_1} A e_{x_2}$ due to the effect of the randomization. We correct for this by adding terms and scaling, leading to the estimator used in Algorithm 1:

$$\widehat{f}_A(\mathcal{R}(x_1), \mathcal{R}(x_2)) = (1-\beta)^{-2}(e_{\mathcal{R}(x_1)} - b)^T A(e_{\mathcal{R}(x_2)} - b).$$

This is an unbiased estimator of the population U-statistic, as for fixed x_1 and x_2 it is an unbiased estimator of $f_A(x_1, x_2)$. Averaging over all pairs of randomized inputs, we get the proposed estimator:

$$\widehat{U}_{f,n} = \binom{n}{2}^{-1} \sum_{1 \le i < j \le n} \widehat{f}_A(\mathcal{R}(x_i), \mathcal{R}(x_j)),$$

which is itself a U-statistic on the randomized sample. As this estimator is unbiased, its mean squared error is equal to its variance, for which the following lemma gives an exact expression.

Lemma 1. *The variance of $\widehat{U}_{f,n}$ is given by*

$$\binom{n}{2}^{-1} \left(\frac{2n-3}{(1-\beta)^2} \mathrm{Var}(e_{(\mathcal{R}(x_1))}{}^T AD) + \frac{1}{(1-\beta)^4} \mathbb{E}(\mathrm{Var}((e_{\mathcal{R}(x_1)} - b) A e_{\mathcal{R}(x_2)} \mid \mathcal{R}(x_1))) \right)$$

Proof. $\widehat{U}_{f,n}$ is a U-statistic, hence its variance is given by (3) where $\zeta_1 = \mathrm{Var}(\mathbb{E}(\widehat{f}_A(\mathcal{R}(x_1), \mathcal{R}(x_2)) \mid \mathcal{R}(x_1)))$ and $\zeta_2 = \mathrm{Var}(\widehat{f}_A(\mathcal{R}(x_1), \mathcal{R}(x_2)))$. We first simplify ζ_1:

$$\begin{aligned}\zeta_1 &= (1-\beta)^{-4} \mathrm{Var}(\mathbb{E}((e_{\mathcal{R}(x_1)} - b)^T A(e_{\mathcal{R}(x_2)} - b) \mid \mathcal{R}(x_1))) \\ &= (1-\beta)^{-4} \mathrm{Var}((e_{\mathcal{R}(x_1)} - b)^T A((1-\beta)D)) \\ &= (1-\beta)^{-2} \mathrm{Var}((e_{\mathcal{R}(x_1)} - b)^T AD) \\ &= (1-\beta)^{-2} \mathrm{Var}((e_{\mathcal{R}(x_1)})^T AD).\end{aligned}$$

Similarly for ζ_2, we have:

$$\begin{aligned}\zeta_2 &= \mathrm{Var}(\mathbb{E}(\widehat{f}(\mathcal{R}(x_1), \mathcal{R}(x_2)) \mid \mathcal{R}(x_1))) + \mathbb{E}(\mathrm{Var}(\widehat{f}(\mathcal{R}(x_1), \mathcal{R}(x_2)) \mid \mathcal{R}(x_1))) \\ &= \zeta_1 + (1-\beta)^{-4} \mathbb{E}(\mathrm{Var}((e_{\mathcal{R}(x_1)} - b)^T A(e_{\mathcal{R}(x_2)} - b) \mid \mathcal{R}(x_1))) \\ &= \zeta_1 + (1-\beta)^{-4} \mathbb{E}(\mathrm{Var}((e_{\mathcal{R}(x_1)} - b)^T A e_{\mathcal{R}(x_2)} \mid \mathcal{R}(x_1))).\end{aligned}$$

Substituting the values of ζ_1 and ζ_2 in the variance expression gives the result. ■

Assuming a uniform bound on the values of f allows a clear and simple bound on the variance.

Corollary 2. *If $f(x, x') \in [0, 1]$ for all x, x', then*

$$\mathrm{Var}(\widehat{U}_{f,n}) \leq \frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4}.$$

Proof. Under the boundedness of f, the random variable $(e^{\mathcal{R}(x_1)})^T AD$ takes values in $[0, 1]$ and so has variance at most $1/4$, whilst the random variable $(e^{\mathcal{R}(x_1)} - B)^T A e^{\mathcal{R}(x_2)}$ takes values in $[-\beta, 1]$ and so has variance at most $(1+\beta)^2/4$. Substituting these into Lemma 1 gives the result. ■

To achieve local differential privacy with parameter ϵ, β should be taken to be $k/(k+e^\epsilon-1)$. This leads directly to the following result.

Corollary 3 (Variance under randomized reponse)**.** *Let $\mathcal{X} = [k]$ and assume f takes values in $[0, 1]$. We have:*

$$\mathrm{Var}(\widehat{U}_{f,n}) \approx \frac{(1+k/\epsilon)^2}{n} + \frac{(1+k/\epsilon)^4}{2n^2} \approx \frac{k^2}{n\epsilon^2},$$

where the approximation holds for small ϵ and $n \gg k^2/\epsilon^2$.

The above result shows that for fixed ϵ the error incurred by this estimator is within a constant factor of the error due to the finite sample setting. As expected, k should be reasonably small for the protocol to yield any utility.

A.2 Bounding the Error of Quantization

We now study the effect of quantization, which is needed to control the error due to privacy when the domain is continuous or has large cardinality. Recall that we quantize $\mathcal{X}$ using a projection $\pi : \mathcal{X} \to [k]$ (assumed to be simple rounding for simplicity), and our goal is to approximate $U_f = \mathbb{E}_{\mu\times\mu}(f(x, y))$ by $U_{A,\pi} = \mathbb{E}_{\mu\times\mu}(A_{\pi(x),\pi(y)})$, which we can privately estimate using the results of Section A.1.

The error incurred by the quantization can be written as follows:

$$\begin{aligned}(U_f - U_{A,\pi})^2 &\leq \int\int (f(x, y) - A_{\pi(x),\pi(y)})^2 d\mu(y) d\mu(x) \\ &= \sum_{i,j=1}^{k} \int_{\pi^{-1}(i)} \int_{\pi^{-1}(j)} (f(x, y) - A_{i,j})^2 d\mu(y) d\mu(x). \qquad (12)\end{aligned}$$

To bound this quantization error, we need additional assumptions. We consider two options, each suggesting a different choice for the quantized kernel $A_{i,j}$. We first consider a Lipschitz assumption on the original kernel function, for which the preferred quantized kernel minimizes the worst-case bound on (12). Then, we consider a smoothness assumption on the data distribution, leading to a quantized kernel that attempts to minimize the average-case error. We stress the fact that in some cases these quantized statistics will match or at least be very close, meaning that the particular choice of quantized kernel will not be crucial.

A.2.1 Assumption of Lipschitz Kernel Function

Our first assumption is motived by the fact that for all data distributions μ, the quantization error (12) can be bounded by

$$\sum_{i,j=1}^{k} \mu(\pi^{-1}(i))\mu(\pi^{-1}(j)) \max_{\substack{x\in\pi^{-1}(i)\\ y\in\pi^{-1}(j)}} (f(x,y) - A_{i,j})^2. \tag{13}$$

This bound is minimized by choosing the quantized kernel to be

$$A^{Mid}_{i,j} = \frac{1}{2} \max_{\substack{x\in\pi^{-1}(i)\\ y\in\pi^{-1}(j)}} f(x,y) + \frac{1}{2} \min_{\substack{x\in\pi^{-1}(i)\\ y\in\pi^{-1}(j)}} f(x,y), \tag{14}$$

which we will call the midpoint kernel. With this kernel we can define

$$\Delta_{i,j} = \frac{1}{4}\Big(\max_{x\in\pi^{-1}(i),y\in\pi^{-1}(j)} f(x,y) - \min_{x\in\pi^{-1}(i),y\in\pi^{-1}(j)} f(x,y)\Big)^2,$$

which allows us to write the bound in equation 13 as

$$\Delta = \sum_{i,j=1}^{k} \mu(\pi^{-1}(i))\mu(\pi^{-1}(j))\Delta_{i,j}.$$

Note that this is itself a discrete U-statistic over the population. The error can now be bounded through a bound on Δ. A natural way to achieve this is to uniformly bound $\Delta_{i,j}$, which can be done by assuming that the kernel function f is Lipschitz. This allows to control the error within each bin.

Lemma 2 (Quantization error for Lipschitz kernel functions)**.** *Let $\mathcal{X} = [0,1]$ and assume $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is L_f-Lipschitz in each input. Let the set of bins be $\{(2l-1)/2k : l \in [k]\}$ and let the quantization scheme π perform simple rounding of inputs (affecting them to the nearest bin). Then we have $(U_f - U_{A,\pi})^2 \leq L_f^2/2k^2$.*

Proof. By the Lipschitz property of f, we have that $|f(x,y)-f(x',y')| \leq L_f(|x-x'|+|y-y'|)$ for all x, x', y, y'. Since the diameter of each bin is equal to $1/k$, we have $\Delta_{i,j} \leq L_f^2/2k^2$ for all $i, j \in [k]$ and the lemma follows. ■

As desired, the quantization error decreases with k. Note that the Lipschitz assumption is met in the important case of the Gini mean difference, while it does not hold for AUC and Kendall's tau. Bounding Δ is not the right approach for such kernels: indeed, for AUC, $\Delta_{i,i} = 1/2$ and so for data distributions μ with $\mu(\pi^{-1}(i)) = 1$ for some i, the quantization error $\Delta \geq 1/2$. In the next section, we consider generic kernel functions under a smoothness assumption on the data distributions.

Remark 3 (Empirical Estimation of Δ)**.** *The quantization error Δ is a discrete U-statistic which can be estimated from the data collected to estimate U. This provides a good empirical estimate of Δ after the fact. However, as the data has to be collected before the estimate can be made it provides no guidance in choosing π (this might be addressed by a multi-round protocol, which we leave for future work). The empirical assessment of Δ may provide a tighter bound on the actual error than can be ascertained by the worst-case Lipschitz assumption.*

A.2.2 Assumption of Smooth Data Distribution

We now consider a smoothness assumption on the data distribution μ. Specifically, we assume that the density $d\mu/d\lambda$ with respect to a measure λ (which varies little on $\pi^{-1}(i)$ for all i) is C-Lipschitz.

In this case, a more sensible choice of quantized kernel is given by

$$A_{i,j}^{Avg} = \frac{1}{\lambda(\pi^{-1}(i))\lambda(\pi^{-1}(j))} \int_{\pi^{-1}(i)} \int_{\pi^{-1}(j)} f(x,y) d\lambda(y) d\lambda(x), \tag{15}$$

which we call the average kernel as the value of $A_{i,j}^{Avg}$ corresponds to the (normalized) expectation of $f(x,y)$, with respect to λ, over points x and y that are mapped to bin i and j respectively.

Under our smoothness assumption, the quantization error (12) can be bounded as follows.

Lemma 3 (Quantization error for smooth distributions)**.** *Let $\mathcal{X} = [0,1]$ and $f(x,y) \in [0,1]$ for all x, y.*[2] *Assume that $d\mu/d\lambda$ is L_μ-Lipschitz. Then we have $(U_f - U_{A,\pi})^2 \leq 4L_\mu^2 D^2(1 + L_\mu^2 D^2)$, where D is the maximum diameter of the quantization bins.*

Proof. For notational convenience, let us denote $\bar{\mu_i} := \mu(\pi^{-1}(i))$ and $\bar{\lambda_i} := \lambda(\pi^{-1}(i))$ for each i. The absolute quantization error with quantized kernel (15) is given by:

$$\begin{aligned}
&\left| \sum_{i,j=1}^{k} \int_{\pi^{-1}(i)} \int_{\pi^{-1}(j)} f(x,y) - A_{i,j}^{Avg} d\mu(y) d\mu(x) \right| \\
\leq &\sum_{i,j=1}^{k} \left| \int_{\pi^{-1}(i)} \int_{\pi^{-1}(j)} f(x,y) - A_{i,j}^{Avg} d\mu(y) d\mu(x) \right| \\
= &\sum_{i,j=1}^{k} \left| \int_{\pi^{-1}(i)} \int_{\pi^{-1}(j)} f(x,y) d\mu(y) d\mu(x) - \bar{\mu_i}\bar{\mu_j} A_{i,j}^{Avg} \right| \qquad (16)
\end{aligned}$$

Note that

$$\int_{\pi^{-1}(i)} \int_{\pi^{-1}(j)} f(x,y) d\mu(y) d\mu(x) = \int_{\pi^{-1}(i)} \int_{\pi^{-1}(j)} f(x,y) \frac{d\mu(y)}{d\lambda(y)} \frac{d\mu(x)}{d\lambda(x)} d\lambda(y) d\lambda(x),$$

and

$$\bar{\mu_i}\bar{\mu_j} A_{i,j}^{Avg} = \frac{\bar{\mu_i}\bar{\mu_j}}{\bar{\lambda_i}\bar{\lambda_j}} \int_{\pi^{-1}(i)} \int_{\pi^{-1}(j)} f(x,y) d\lambda(y) d\lambda(x).$$

[2] Similar arguments can be made in more general metric spaces.

Plugging these equations into (16) we get:

$$
\begin{aligned}
&\left|\int_{\pi^{-1}(i)}\int_{\pi^{-1}(j)} f(x,y)d\mu(y)d\mu(x) - \bar{\mu}_i\bar{\mu}_j A^{Avg}_{i,j}\right| \\
=&\left|\int_{\pi^{-1}(i)}\int_{\pi^{-1}(j)} f(x,y)\Big(\frac{d\mu(y)}{d\lambda(y)}\frac{d\mu(x)}{d\lambda(x)} - \frac{\bar{\mu}_i\bar{\mu}_j}{\bar{\lambda}_i\bar{\lambda}_j}\Big)d\lambda(y)d\lambda(x)\right| \\
\leq&\int_{\pi^{-1}(i)}\int_{\pi^{-1}(j)}\left|\frac{d\mu(y)}{d\lambda(y)}\frac{d\mu(x)}{d\lambda(x)} - \frac{\bar{\mu}_i\bar{\mu}_j}{\bar{\lambda}_i\bar{\lambda}_j}\right| d\lambda(y)d\lambda(x) \qquad (17)\\
\leq&\int_{\pi^{-1}(i)}\int_{\pi^{-1}(j)} L_\mu D\Big(\max_{z\in\pi^{-1}(i)} d\mu(z)/d\lambda(z) + \max_{w\in\pi^{-1}(j)} d\mu(w)d\lambda(w)\Big)d\lambda(y)d\lambda(x) \qquad (18)\\
\leq&\int_{\pi^{-1}(i)}\int_{\pi^{-1}(j)}(L_\mu D)d\lambda(y)d\mu(x) + \int_{\pi^{-1}(i)}\int_{\pi^{-1}(j)}(L_\mu D)d\mu(y)d\lambda(x) \\
&+\int_{\pi^{-1}(i)}\int_{\pi^{-1}(j)}(L_\mu D)(2L_\mu D)d\lambda(y)d\lambda(x). \qquad (19)
\end{aligned}
$$

Summing over all i, j and taking the square finally gives the result:

$$
\begin{aligned}
&\left(\int\int f(x,y)d\mu(y)d\mu(x) - \sum \mu(\pi^{-1}(i))\mu(\pi^{-1}(j))A^{Avg}_{i,j}\right)^2 \\
\leq&4L_\mu^2 D^2 + 4L_\mu^4 D^4.
\end{aligned}
$$

■

The diameter of quantization bins is typically of order $1/k$, hence the quantization error is of order $1/k^2$. In practice, λ can simply be taken to be Lebesgue measure, hence computing (15) amounts to averaging the kernel function over all possible points $(x, y) \in \mathcal{X}$ that fall in the bins (i, j), and can be easily approximated by Monte Carlo sampling when one does not have a closed form expression for the integral.

B Details and Proofs for AUC Protocol

B.1 Proof of Theorem 3

We define $R^m = \{p \in \{0,1\}^m : \forall p' \preceq p, \tilde{h}^-_{p'}\tilde{h}^+_{p'} > \tau\}$ as the set of nodes recursed on at level m. Similarly, and for $m > 0$, let $A^m = R^{m-1} \cdot \{0,1\}$ be the active nodes at level m, i.e. those to be either recursed on or discarded. Then, the set of discarded nodes at level m is defined as $D^m = A^m \setminus R^m$. Our algorithm has two main sources of error: (i) the one incurred on by discarded nodes, i.e. nodes in $\bigcup_{i\in[\alpha]} D^m$ for whose intervals the algorithm uses a rough estimate, and (ii) the error in the estimating the contribution to the UAUC of the recursed nodes, i.e. nodes in $\bigcup_{i\in[\alpha]} R^m$.

The threshold τ is carefully chosen according to the error of the estimator $\hat{h}$ to balance these two errors. In this way we translate error bounds for $\hat{h}^+_p, \hat{h}^-_p$ into error bounds for $\widehat{\texttt{UAUC}}$. Our proof starts by bounding the expected size of R^m.

Lemma 4. *Consider the instantiation of Equation 10 with a frequency oracle for estimating $h^{\pm}$ satisfying $\forall p \in \{0,1\}^{\leq\alpha} : \big(\mathbb{E}(\hat{h}_p^{\pm}) = h_p^{\pm}, \mathbb{E}((\hat{h}_p^{\pm} - h_p^{\pm})^2) \leq v^{\pm}\big)$, with $v^{\pm} = Cn^{\pm}\alpha$, i.e. the estimate is unbiased and has uniformly bounded* `MSE`*. If $a > 1$, then for all $m \in [\alpha]$,*

$$\mathbb{E}(|R^m|) \leq \frac{\sqrt{n^+} + \sqrt{n^-}}{2(\sqrt{a}-1)\sqrt{C\alpha}} \leq \frac{1}{\sqrt{a}-1}\sqrt{\frac{n}{2C\alpha}}$$

Proof. Let $\hat{n}^{\pm} = \sum_{p \in A^m} \max(\hat{h}_p^{\pm}, 0)$, the sum of the positive estimated counts of active nodes at level m.

Note that if $p \in R^m$ then $\tilde{h}_p^+ \tilde{h}_p^- \geq a\sqrt{v^+ v^-}$. In this case either, $\hat{h}_p^{\pm} = \tilde{h}_p^{\pm}$ and thus $\hat{h}_p^+ \hat{h}_p^- \geq a\sqrt{v^+ v^-}$, $\hat{h}_p^- \neq \tilde{h}_p^-$ and thus $\hat{h}_p^+ > 2\sqrt{av^+}$, or $\hat{h}_p^- \neq \tilde{h}_p^-$ and thus $\hat{h}_p^- > 2\sqrt{av^-}$. In any of these cases $\hat{h}_p^+/\sqrt{av^+} + \hat{h}_p^-/\sqrt{av^-} \geq 2$.

Therefore

$$2|R^m| \leq \sum_{p \in R^m} \frac{\hat{h}_p^+}{\sqrt{av^+}} + \frac{\hat{h}_p^-}{\sqrt{av^-}} \leq \frac{\hat{n}^+}{\sqrt{av^+}} + \frac{\hat{n}^-}{\sqrt{av^-}}$$

and thus

$$\mathbb{E}(|R^m|) \leq \frac{\mathbb{E}(\hat{n}^+)}{2\sqrt{av^+}} + \frac{\mathbb{E}(\hat{n}^-)}{2\sqrt{av^-}}.$$

We bound $\mathbb{E}(\hat{n}^{\pm})$ as follows

$$\begin{aligned}
\mathbb{E}(\hat{n}^{\pm}) = \sum_{p \in A^m} \mathbb{E}(\max(\hat{h}_p^{\pm}, 0)) &\leq n^{\pm} + \sum_{p \in A^m} \mathbb{E}(\max(e_p^{\pm}, 0)) \\
&\leq n^{\pm} + \mathbb{E}(|A^m|) \max_{p \in A^m} \mathbb{E}(\max(e_p^{\pm}, 0)) \leq n^{\pm} + \mathbb{E}(|R^{m-1}|) \max_{p \in A^m} \mathbb{E}(|e_p^{\pm}|) \\
&\leq n^{\pm} + \mathbb{E}(|R^{m-1}|) \max_{p \in A^m} \sqrt{\mathbb{E}(|e_p^{\pm}|^2)} \leq n^{\pm} + \mathbb{E}(|R^{m-1}|)\sqrt{v_{\pm}}
\end{aligned}$$

We can now use this to bound the expression for $\mathbb{E}(|R^m|)$.

$$\mathbb{E}(|R^m|) \leq \frac{n^+}{2\sqrt{av^+}} + \frac{n^-}{2\sqrt{av^-}} + \frac{\mathbb{E}(|R^{m-1}|)}{\sqrt{a}} \tag{20}$$

We now need a bound on $\mathbb{E}(|R^{m-1}|)$ so we will proceed by induction.

Let $B = \frac{\sqrt{n^+}+\sqrt{n^-}}{2(\sqrt{a}-1)\sqrt{C\alpha}}$. We take $\mathbb{E}(|R^{m-1}|) \leq B$ as the induction hypothesis, and $\mathbb{E}(|R^0|) = 1 \leq B$ as the base case.

The expression on the right hand side of inequality 20 is a monotonically increasing function of $\mathbb{E}(|R^{m-1}|)$ and has a fixed point

$$\frac{\frac{n^+}{2\sqrt{av^+}} + \frac{n^-}{2\sqrt{av^-}}}{1 - \frac{1}{\sqrt{a}}} = \frac{\frac{n^+}{2\sqrt{v^+}} + \frac{n^-}{2\sqrt{v^-}}}{\sqrt{a}-1} = \frac{\sqrt{\frac{n^+}{C\alpha}} + \sqrt{\frac{n^-}{C\alpha}}}{2(\sqrt{a}-1)} = \frac{\sqrt{n^+} + \sqrt{n^-}}{2(\sqrt{a}-1)\sqrt{C\alpha}} = B.$$

Thus we can conclude that

$$\mathbb{E}(|R^m|) \leq B \tag{21}$$

completing the induction and thus (21) holds for all m.

Finally we note that

$$\sqrt{n^+} + \sqrt{n^-} \leq \sqrt{2n}$$

and so

$$\frac{\sqrt{n^+} + \sqrt{n^-}}{2(\sqrt{a}-1)\sqrt{C\alpha}} \leq \frac{1}{\sqrt{a}-1}\sqrt{\frac{n}{2C\alpha}}$$

completing the proof. ■

We are now ready to prove Theorem 3.

Proof of Theorem 3. The estimation error $E_p = \widehat{\texttt{UAUC}}(\hat{h}_p^+, \hat{h}_p^-) - \texttt{UAUC}(h_p^+, h_p^-)$ at a given node p can be written recursively as follows:

$$E_p = \begin{cases} \frac{1}{2}(\hat{h}_{p\cdot 1}^+ + \hat{h}_{p\cdot 0}^+)(\hat{h}_{p\cdot 1}^- + \hat{h}_{p\cdot 0}^-) - \texttt{UAUC}(h_p^+, h_p^-) & \text{if } p \in D \\ 0 & \text{if } p \text{ is a leaf} \\ \hat{h}_{p\cdot 1}^+ \hat{h}_{p\cdot 0}^- - h_{p\cdot 1}^+ h_{p\cdot 0}^- + E_{p\cdot 0} + E_{p\cdot 1} & \text{if } p \in R \end{cases}$$

We will consider the error E_λ in two parts. Firstly, there is the contribution E^D from those prefixes $p \in D$ which we define by setting

$$E_m^D = \sum_{p \in D_{m-1}} \frac{1}{2}(\hat{h}_{p\cdot 1}^+ + \hat{h}_{p\cdot 0}^+)(\hat{h}_{p\cdot 1}^- + \hat{h}_{p\cdot 0}^-) - \texttt{UAUC}(h_p^+, h_p^-)$$

and $E^D = \sum_{m\in[\alpha]} E_m^D$. Secondly, there is the contribution from the prefixes $p \in R$ excluding their recursive subcalls which we define by setting

$$E_m^R = \sum_{p \in R_{m-1}} \hat{h}_{p\cdot 1}^+ \hat{h}_{p\cdot 0}^- - h_{p\cdot 1}^+ h_{p\cdot 0}^-$$

and $E^R = \sum_{m\in[\alpha]} E_m^R$. In bounding both of these we will make use of conditioning on $\mathcal{F}_m = (\hat{h}_p^-, \hat{h}_p^+)_{p\in\{0,1\}^{\leq m}}$ i.e. the answers of the frequency oracles for layers up to m.

We start by bounding E^R. For any $m \in [\alpha]$, we first show that E_m^R is a martingale difference sequence i.e.

$$\begin{aligned} \mathbb{E}(E_m^R|\mathcal{F}_{m-1}) &= \mathbb{E}(\sum_{p\in R^{m-1}} (\hat{h}_{p\cdot 1}^+ \hat{h}_{p\cdot 0}^- - h_{p\cdot 1}^+ h_{p\cdot 0}^-)|\mathcal{F}_{m-1}) \\ &= \sum_{p\in R^{m-1}} \mathbb{E}((h_{p\cdot 1}^+ + e_{p\cdot 1}^+)(h_{p\cdot 0}^- + e_{p\cdot 0}^-) - h_{p\cdot 1}^+ h_{p\cdot 0}^-) \\ &= \sum_{p\in R^{m-1}} \mathbb{E}(h_{p\cdot 1}^+ e_{p\cdot 0}^- + e_{p\cdot 0}^+ h_{p\cdot 0}^- + e_{p\cdot 1}^+ e_{p\cdot 0}^-) \\ &= 0 \end{aligned}$$

where the final equality holds because $\mathbb{E}(e_p^\pm) = 0$ for all p and $e_{p\cdot 1}^+$ and $e_{p\cdot 0}^-$ are independent. From this we can conclude that for $m' > m$

$$\mathbb{E}(E_m^R E_{m'}^R) = \mathbb{E}(\mathbb{E}(E_m^R E_{m'}^R|\mathcal{F}_{m'-1}))) = \mathbb{E}(E_m^R \mathbb{E}(E_{m'}^R|\mathcal{F}_{m'-1})) = \mathbb{E}(0) = 0$$

and thus

$$\mathbb{E}(E^{R^2}) = \mathbb{E}(\sum_{m\in[\alpha]}\sum_{m'\in[\alpha]} E^R_m E^R_{m'}) = \sum_{m\in[\alpha]} \mathbb{E}(E^{R^2}_m) = \sum_{m\in[\alpha]} \mathbb{E}(\mathbb{E}(E^{R^2}_m|\mathcal{F}_{m-1})). \tag{22}$$

Next we shall bound $\mathbb{E}(E^{R^2}_m|\mathcal{F}_{m-1})$. We start by writing out

$$\mathbb{E}(E^{R^2}_m|\mathcal{F}_{m-1}) = \mathbb{E}((\sum_{p\in R^{m-1}} (\hat{h}^+_{p\cdot 1}\hat{h}^-_{p\cdot 0} - h^+_{p\cdot 1}h^-_{p\cdot 0}))^2|\mathcal{F}_{m-1}).$$

By Equation 22 this becomes

$$\mathbb{E}(E^{R^2}_m|\mathcal{F}_{m-1}) = \sum_{p\in R^{m-1}} \mathbb{E}((\hat{h}^+_{p\cdot 1}\hat{h}^-_{p\cdot 0} - h^+_{p\cdot 1}h^-_{p\cdot 0})^2|\mathcal{F}_{m-1}).$$

After expanding the above and removing all the terms that are zero, because they are the expected value of the product of $e^{\pm}_{p\cdot i}$ with something independent of it, we are left with

$$\begin{aligned}
\mathbb{E}(E^{R^2}_m|\mathcal{F}_{m-1}) &= \sum_{p\in R^{m-1}} \mathbb{E}({h^+_{p\cdot 1}}^2{e^-_{p\cdot 0}}^2 + {e^+_{p\cdot 0}}^2{h^-_{p\cdot 0}}^2 + {e^+_{p\cdot 1}}^2{e^-_{p\cdot 0}}^2) \\
&\leq \sum_{p\in R^{m-1}} (v^-{h^+_{p\cdot 1}}^2 + v^+{h^-_{p\cdot 0}}^2) + |R^m|v^+v^- \\
&\leq v^-{n^+}^2 + v^+{n^-}^2 + |R^m|v^+v^-.
\end{aligned}$$

Subbing this into (22) and using Lemma 4 gives

$$\begin{aligned}
\mathbb{E}(E^{R^2}) &\leq \alpha \max_m \mathbb{E}(v^-{n^+}^2 + v^+{n^-}^2 + |R^m|v^+v^-) \\
&= \alpha(v^-{n^+}^2 + v^+{n^-}^2 + \max_m \mathbb{E}(|R^m|)v^+v^-) \\
&\leq n^+n^-C\alpha^2(n^+ + n^- + \frac{C\alpha}{\sqrt{a}-1}\sqrt{\frac{n}{2C\alpha}}) \\
&\leq n^+n^-C\alpha^2(n + \frac{\sqrt{C\alpha n}}{\sqrt{2}(\sqrt{a}-1)}) \\
&=: B^R
\end{aligned}$$

To bound E^D, first define $E^F_m = \sum_{p\in D_{m-1}} \frac{1}{2}(\hat{h}^+_{p\cdot 1} + \hat{h}^+_{p\cdot 0})(\hat{h}^-_{p\cdot 1} + \hat{h}^-_{p\cdot 0}) - \frac{1}{2}h^+_p h^-_p$ and $E^G_m = \sum_{p\in D_m} \frac{1}{2}h^+_p h^-_p - \texttt{UAUC}(h^+_p, h^-_p)$. We refer to the leaves in $[0..2^\alpha - 1]$ *covered* by a path p as $\mathcal{I}(p) = \{i \in [0..d-1] : p \preceq b_i\}$. Now note that

$$E^D = \sum_{p\in D} \frac{1}{2}(\hat{h}^+_{p\cdot 1} + \hat{h}^+_{p\cdot 0})(\hat{h}^-_{p\cdot 1} + \hat{h}^-_{p\cdot 0}) - \sum_{i\in\mathcal{I}(p)} h^+_{b_i} \sum_{j\in\mathcal{I}(p), j<i} h^-_{b_j} = \sum_{m\in[\alpha]} E^F_m + \sum_{m\in[\alpha]} E^G_m.$$

We now bound E_m^F and E_m^G separately. For a leaf node s, let us denote by $v(s)$ the *unique* node in D that is a prefix of s. We then have:

$$\begin{aligned}\mathbb{E}((\sum_{m\in[\alpha]} E_m^G)^2) &= \mathbb{E}((\sum_{p\in D}\frac{1}{2}h_p^+h_p^- - \texttt{UAUC}(h_p^+,h_p^-))^2)\\ &\le \mathbb{E}((\sum_{p\in D}\frac{1}{2}h_p^+h_p^-)^2) = \frac{1}{4}\mathbb{E}((\sum_{p\in D}\sum_{i\in\mathcal{I}(p)} h_{b_i}^+ \sum_{j\in\mathcal{I}(p),j<i} h_{b_j}^-)^2)\\ &\le \frac{{n^+}^2}{4}\max_{s\in[0..d-1]}\mathbb{E}((h_{v(s)}^-)^2).\end{aligned}$$

We can then bound

$$\begin{aligned}\mathbb{E}({h_{v(s)}^-}^2) &= \sum_{p\preceq p(s)}\mathbb{E}({h_p^-}^2\mathbb{I}_{p=v(s)}) = \sum_{p\preceq p(s)}\mathbb{E}((\hat{h}_p^- - e_p^-)^2\mathbb{I}_{p=v(s)})\\ &\le \sum_{p\preceq s}\mathbb{E}((2\sqrt{av^-} - e_p^-)^2\mathbb{I}_{p=v(s)}) \le \sum_{p\preceq s}\mathbb{E}(4av^- - 4\sqrt{av^-}e_p^- + {e_p^-}^2)\\ &\le (4a+1)\alpha v^-.\end{aligned}$$

Thus

$$\mathbb{E}({E^G}^2) \le {n^+}^2(a+1/4)\alpha v^- \le C(a+1/4)n^-{n^+}^2\alpha^2.$$

Furthermore by symmetry between $-$ and $+$

$$\mathbb{E}({E^G}^2) \le C(a+1/4)n^-n^+\min(n^-,n^+)\alpha^2 =: B^G.$$

Secondly we bound $\sum_{m\in[\alpha]} E_m^F$. Note that E_{m-1}^F is a function of $\mathcal{F}_{m-1}$ and $\mathbb{E}(E_m^F|\mathcal{F}_{m-1}) = 0$ so

$$\begin{aligned}\mathbb{E}((\sum_m E_m^F)^2) &= \mathbb{E}(\sum_m {E_m^F}^2) = \sum_m \mathbb{E}({E_m^F}^2) = \sum_m \mathbb{E}(\mathbb{E}({E_m^F}^2|\mathcal{F}_{m-1}))\\ &\le \sum_m \mathbb{E}(\mathbb{E}((\sum_{p\in D_{m-1}}\frac{1}{2}(\hat{h}_{p\cdot 1}^+ + \hat{h}_{p\cdot 0}^+)(\hat{h}_{p\cdot 1}^- + \hat{h}_{p\cdot 0}^-) - \frac{1}{2}h_p^+h_p^-)^2|\mathcal{F}_{m-1}))\end{aligned}$$

Similarly to the bound on E^R, we now apply the pairwise independence property and note that $\hat{h}_{p\cdot 1}^\pm + \hat{h}_{p\cdot 0}^\pm$ is an unbiased estimator of $h_p^\pm$ with variance bounded by $2v^\pm$. This results in

$$\begin{aligned}\mathbb{E}((\sum_m E_m^F)^2) &\le \sum_m \mathbb{E}(\sum_{p\in A_{m-1}}\mathbb{I}_{\tilde{h}_p^+\tilde{h}_p^-<\tau}\mathbb{E}({h_p^+}^2v^-/2 + v^+{h_p^-}^2/2 + v^+v^-|\mathcal{F}_{m-1}))\\ &\le \sum_m v^-\mathbb{E}(\sum_{p\in A_{m-1}}\mathbb{I}_{\tilde{h}_p^+\tilde{h}_p^-<\tau}{h_p^+}^2)/2 + v^+\mathbb{E}(\sum_{p\in A_{m-1}}\mathbb{I}_{\tilde{h}_p^+\tilde{h}_p^-<\tau}{h_p^-}^2)/2 + v^+v^-\mathbb{E}(|A_{m-1}|).\end{aligned}$$

Noting that $\mathbb{E}(\mathbb{I}_{\tilde{h}_p^+\tilde{h}_p^-<\tau}{h_p^+}^2) \le \min({h_p^+}^2, \frac{{h_p^+}^2v^+}{(h_p^+-\sqrt{v^+})^2}) \le 4v^+$ and that $\mathbb{E}(|A_{m-1}|) = 2\mathbb{E}(|R_{m-2}|) \le \frac{1}{\sqrt{a}-1}\sqrt{\frac{n}{2C\alpha}}$ gives

$$\mathbb{E}((\sum_m E_m^F)^2) \le \sum_m E(|A_{m-1}|)5v^+v^- \le \frac{5\sqrt{2n}C^{1.5}\alpha^{2.5}n^+n^-}{\sqrt{a}-1} := B^F.$$

By the Cauchy-Schwarz inequality we can conclude that,

$$\mathbb{E}(E^{D^2}) \leq 2(B^G + B^F).$$

Finally applying Cauchy-Schwarz again gives

$$\begin{aligned}\mathbb{E}(E_\lambda^2) &= \mathbb{E}((E^R + E^G + E^F)^2) \\ &\leq 2B^R + 4B^G + 4B^F \\ &= Cn^-n^+\alpha^2(2n + (4a+1)\min(n^-, n^+) + \frac{21\sqrt{2nC\alpha}}{\sqrt{a}-1})\end{aligned}$$

■

Remark 4. *The use of Cauchy-Schwarz to combine the separate errors in this proof is optimized for simplicity rather than minimizing the constants. At the expense of making the bound substantially more complicated a more precise analysis would reduce the bound. Gaining up to a factor of two in the case of very large n and $min(n^-, n^+)$ small compared to n.*

The value of a in Theorem 3 can be chosen to minimize the error by taking it to solve $\sqrt{a}(\sqrt{a}-1)^2 = 21\sqrt{2nc\alpha}/(8\min(n^-, n^+))$ which is approximately

$$a = (1 + \sqrt{21/8}(2Cn\alpha/\min(n^-, n^+)^2)^{\frac{1}{4}})^2.$$

This leads to the following corollary.

Corollary 4. *Let $n_{\min} = \min(n^-, n^+)$ and $a = (1 + \sqrt{21/8}(2Cn\alpha/n_{\min}^2)^{\frac{1}{4}})^2 = 1 + o(1)$ then*

$$\mathit{MSE}(\widehat{\mathit{AUC}}) \leq \frac{C}{n_{\min}(n - n_{\min})}\alpha^2(2n + (4a+1)n_{\min} + 14(Cnn_{\min}^2\alpha)^{\frac{1}{4}}) = O(\alpha^2/n_{\min}).$$

Remark 5. *For fixed α this is of the same order as the sampling error incurred in non-private AUC.*

Algorithm variant. An alternative algorithm assigns a value of zero to edges that it discards. For this algorithm a similar theorem holds by the same argument (actually a slightly simpler argument) the resulting error bound is

$$\mathtt{MSE}(\widehat{\mathtt{UAUC}}) = Cn^-n^+\alpha^2(2n + (8a+2)\min(n^-, n^+) + \frac{\sqrt{2C\alpha n}}{(\sqrt{a}-1)}).$$

Note that the second term which is of leading order for $\min(n^-, n^+)$ a fixed fraction of n is twice as large however the final term which is lower order is twenty-one times smaller. This lower order term might not be negligible in practice and so this algorithm should be considered. The corresponding choice of a and bound on the final error is given by the following result.

Corollary 5. *Let $n_{\min} = \min(n^-, n^+)$ and $a = (1 + \sqrt{\frac{\sqrt{2C\alpha n}}{16n_{\min}}})^2 = 1 + o(1)$ then*

$$\mathit{MSE}(\widehat{\mathit{AUC}}) \leq \frac{C}{n_{\min}(n - n_{\min})}\alpha^2((8a+2)n_{\min} + 2n + (512C\alpha n n_{\min}^2)^{\frac{1}{4}}) = O(\alpha^2/n_{\min})$$

Algorithm 2: Local Randomizer

Public Parameters: Domain size 2^l, privacy budget ϵ.
Input: Private index q
Output: A single bit z submitted to the server

$j \leftarrow [0..2^l - 1]$ ▷ `Selected uniformly at random`
$y := \frac{1}{\sqrt{2^l}}(-1)^{\langle j,q \rangle}$ ▷ y `is` M_{j,x^l}`, where` $M \in \{-1,1\}^{2^l \times 2^l}$ `is a Hadamard matrix`
$z := \begin{cases} y & \text{with probability } \frac{e^\epsilon}{1+e^\epsilon} \\ \neg y & \text{otherwise} \end{cases}$ ▷ `Submit randomized response on` y
Send j, z to the Aggregator

B.2 Instantiating the Private Hierarchical Histogram $\hat{h}$

Theorem 3 does not yield a complete algorithm as it only states that, if we had a differentially private algorithm for computing estimates of a hierarchical histogram that satisfy the conditions of Theorem 3, then we could solve AUC with the stated accuracy. In this section we instantiate such algorithm and show that, besides the required error guarantees, our proposal also has other nice properties, namely (i) it is one round, (ii) each user sends a single bit, and (iii) it is sublinear in d processing space at the server.

B.2.1 Frequency Oracle

Relevant previous work on estimating hierarchical histograms in the local model includes the work of Bassily et al. (2017). While in that work the target problem is heavy hitters, their algorithm is similar to ours, as the server retrieves the heavy hitters by exploring a hierarchical histogram. Moreover their protocol – called `TreeHist` – has the nice properties listed above, as it is one round, every user sends a single bit and requires reconstruction space sublinear in d. This satisfies the three above conditions. It is thus tempting to reuse the hierarchical histogram construction from Bassily et al. (2017). However, it does not satisfy the conditions of Theorem 3, as it is not guaranteed to be unbiased.

Alternative recent algorithms for constructing hierarchical histograms in the local model are presented in Kulkarni et al. (2019), with the motivation of answering range queries over a large domain. This proposal is much closer to what we need. However, it has some shortcomings: first, although it is one round, each user sends $O(\log(d))$ bits, and more importantly, it requires space $O(d)$ space at the server, as it reconstructs the whole hierarchical histogram. However, one can tweak the protocol from Kulkarni et al. (2019) to overcome these limitations. We shall first split the users into $\log(d)$ groups (one for each level) and then for each level we shall apply the frequency oracle. Algorithm 2 and Algorithm 3 show the local randomizer (user side) and frequency oracle (server side) for each histogram.

Let count(q) be the true count of an index q in a histogram. The following lemma is shown in Kulkarni et al. (2019).

Lemma 5. *The frequency oracle, Algorithm 3, run with n_l users is unbiased* $\mathbb{E}(z_q) = count(q)$

Algorithm 3: Frequency Oracle

Public Parameters: Domain size 2^l, privacy budget ϵ.
Input: The index j_i and response z_i of each party i and an index q to estimate the frequency of
Output: z an estimated count of q

1 For all i, $y_i := \frac{1}{\sqrt{2^l}}(-1)^{\langle j_i, q\rangle}$ ▷ y_i is $M_{j_i,q}$, where $M \in \{-1,1\}^{2^l \times 2^l}$ is a Hadamard matrix
2 $z_q := \frac{e^\epsilon+1}{e^\epsilon-1}\sum_i y_i z_i$ ▷ De-bias the sum of contributions
3 Return z_q

and satisfies the following bound on the `MSE`*.*

$$\mathbb{E}((z_q - count(q))^2) \leq \frac{4n_l e^\epsilon}{(e^\epsilon - 1)^2} \tag{23}$$

Additionally we require the following lemma on the frequency oracle satisfies condition (3) in Theorem 3 which is given by the following lemma.

Lemma 6. *For distinct $q, q' \in [0..2^l - 1]$, z_q and $z_{q'}$ are independent i.e. the responses of the oracle are pairwise independent.*

Proof. As each user is independent of every other user it suffices to show that each user's contribution to the two entries are independent. Suppose that a user has input $q'' \neq q'$, chooses index j to report and let b be a bit indicating that the user chose $z = \neg y$ in Algorithm 2. That user's contributions to the two estimates (scaled by 2^l) are $(-1)^{\langle j,q\rangle + \langle j,q''\rangle + b}$ and $(-1)^{\langle j,q'\rangle + \langle j,q''\rangle + b}$. Note that we can consider j, q, q' and q'' as elements of $\mathbb{F}_2^l$. Then $q + q''$ and $q' + q''$ are distinct and $q' + q'' \neq 0$. These two facts imply respectively that $\langle j, q' + q''\rangle$ is independent of $\langle j, q + q''\rangle$ and that $\langle j, q' + q''\rangle$ is uniformly distributed in $\mathbb{F}_2$. Thus the contributions are independent. ■

B.2.2 Splitting Strategies

We will instantiate $\hat{h}$ by running the frequency oracle above for each level of the hierarchy. The main choice remaining is how to determine which users contribute to each layer, we will consider two possibilities here. Firstly we can have everyone contribute to all layers, splitting their privacy budget. Alternatively, users can be split evenly across levels at random, each contributing to only one frequency oracle. Another possibility is to assign each user to a level independently and uniformly, this is similar to splitting them evenly though adds slightly more noise and is more complicated to analyse. In all cases, conditions 1 and 3 in Theorem 3 follow from Lemmas 5 and 6.

Splitting Privacy Budget Across Layers. In the case of everyone contributing to all layers the privacy budget can be split using either basic or advanced composition. In either case condition 4 from Theorem 3 holds as the randomness for each layer is independent.

For pure differential privacy we must use basic composition. This allows us to run each frequency oracle can be run with a privacy budget of $\tilde{\epsilon} = \epsilon/\alpha$. Lemma 5 then gives a bound of $O_\epsilon(n\alpha^2)$ on the mean squared error of each entry. While this is insufficient to establish condition 2 of Theorem 3, similar arguments can be used to prove that the algorithm built in this way achieves pure differential privacy at the cost of an α factor in the $\texttt{MSE}$.

If we instead settle for (ϵ, δ)-differential privacy, and assume for convenience that $\epsilon \leq \sqrt{\alpha}\ln(2)$, advanced composition allows each frequency oracle to be run with privacy budget $\tilde{\epsilon} = \epsilon/(\sqrt{\alpha}(1 + \sqrt{2\log(1/\delta)}))$. Condition 2 in Theorem 3 then holds for some C depending on ϵ and δ. This is the implementation and analysis that gives Theorem 3 as it is stated.

Splitting Users Across Layers. When splitting users across levels the frequency oracles can each be run with privacy budget ϵ. However, each oracle will have only n/α users and there is a subsampling error between the total sample and the input given to the frequency oracle. The squared error due to subsampling is $O(1/n)$ thus Lemma 5 provides a $O_\epsilon(n\alpha)$ bound on the $\texttt{MSE}$. This means that condition 2 of Theorem 3 holds. This would provide a version of Theorem 3 with pure differential privacy, however condition 4 from Theorem 3 fails to hold. Intuitively this is because if a user contributes to one level they can't contribute to another level. There are still two things that can be proved about this version of the algorithm.

Firstly, it is still possible to prove a result like Theorem 3, but in which the $\texttt{MSE}$ is α times bigger. The proof of this result follows the same steps as that of Theorem 3 except that the martingale difference sequences argument must be replaced by a bound not assuming pairwise independence.

A second way of viewing this algorithm is to think of each input as being drawn independently from some population distribution and then compare the output to the AUC of that distribution. That is, given a pair of distributions $\mathcal{D}^\pm$, $\mathcal{S}^\pm$ is obtained by sampling each value independently from $\mathcal{D}^\pm$. Denote $\texttt{AUC}_{\text{pop}} = \mathbb{E}_{x^+\sim\mathcal{D}^+, x^-\sim\mathcal{D}^-}[f(x^+, x^-)]$ and let $\texttt{MSE}_{\text{pop}}(\widehat{\texttt{AUC}}) = \mathbb{E}[(\texttt{AUC}_{\text{pop}} - \widehat{\texttt{AUC}})^2]$. The fact that each of the users has an independent identically distributed input means that the contribution to each layer is independent, i.e. we can recover Theorem 3 with $\texttt{MSE}$ replaced by $\texttt{MSE}_{\text{pop}}$. This alternative notion of $\texttt{MSE}_{\text{pop}}$ is the correct notion to work with if the purpose of the deployment of the algorithm is to find the AUC of the population the sample is drawn from rather than just of the sample. This is likely to be the case in many applications.

Summary. Table 1 summarizes the choices in the algorithm and analysis. The resulting orders of the $\texttt{MSE}$, corresponding to Theorem 4 are given in the final column.

B.3 Additional AUC experiments

Figure 5 shows the error that our algorithm incurs on two synthetic datasets. The first data set "bench=auc_one" consists of two distinct inputs $(d-1, 1)$ and $(0, -1)$ each occurring 10^6 times. On this data set the algorithm incurs considerable error due to significant recursion error, E_m^R, being incurred for every level. This example illustrates that our analysis is not far from being tight.

Splitting	Analysis	Error in $\widehat{\texttt{AUC}}$
Privacy budget	Basic composition	$\texttt{MSE} \leq O(\frac{\alpha^3}{n\epsilon^2})$
Privacy budget	Advanced composition	$\texttt{MSE} \leq O(\frac{\alpha^2 \log(1/\delta)}{n\epsilon^2})$
Users	w.r.t. sample	$\texttt{MSE} \leq O(\frac{\alpha^3}{n\epsilon^2})$
Users	w.r.t. population	$\texttt{MSE}_{\text{pop}} \leq O(\frac{\alpha^2}{n\epsilon^2})$

Table 1: Summary of error bounds for our AUC protocol for different splitting strategies and analysis techniques.

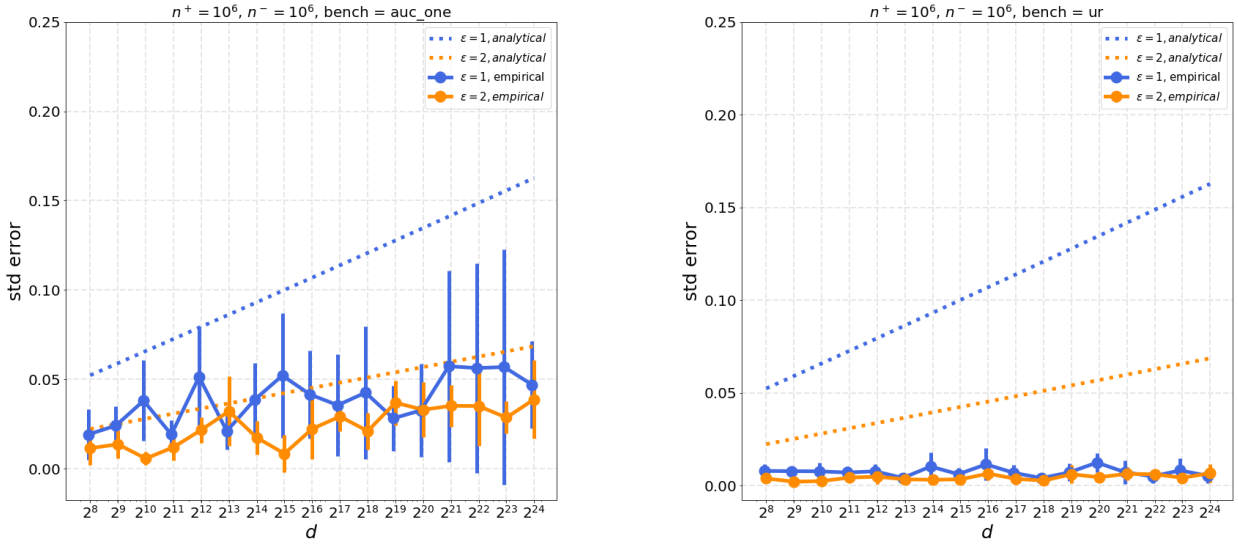

Figure 5: Error incurred by our AUC protocol on two synthetic datasets.

In the second synthetic dataset "bench=ur" there are again 10^6 entries in the positive class and the negative class, however here the s_i are drawn independently and uniformly from $[0..d-1]$. Here the error is much lower, this is because (i) the algorithm does not explore any of the lower sections of the tree and so no recursion error is incurred whilst exploring it and (ii) within intervals that are discarded the points are uniformly distributed so the estimation of the AUC within that interval as a half is effective. Both of these effects will occur approximately whenever the data is smooth so one can expect the algorithm to do better in the case of smooth data than the analytic bounds indicate.

C Details and Proofs for 2PC Protocol

C.1 Proof of Theorem 5 and Discussion

Proof. The ϵ-DP follows from the Laplace mechanism and the simple composition property of DP (observing that each input x_i appears in exactly P pairs in $\mathcal{P}$).

It is easy to see that $\widehat{U}_{f,n}$ is unbiased, hence we only need to bound its variance. We will separate the part due to subsampling and the part due to privacy. To this end, we decompose

$\widehat{U}_{f,n}$ into a noise-free term and a noisy term:

$$\widehat{U}_{f,n} = \underbrace{\frac{2}{Pn}\sum_{(i,j)\in\mathcal{P}} f(x_i, x_j)}_{\widehat{U}_{f,n,P}} + \frac{2}{Pn}\sum_{(i,j)\in\mathcal{P}} \eta_{ij}. \tag{24}$$

The noisy term is an average of independent Laplace random variables: its variance is equal to $2P/n\epsilon^2$.

The quantity $\widehat{U}_{f,n,P}$ is known as an incomplete U-statistic, whose variance is given by Blom (1976):

$$\mathrm{Var}(\widehat{U}_{f,n,P}) = \frac{4}{(Pn)^2}\Big(\mathbb{E}[f_1(\mathcal{P})]\zeta_1 + \mathbb{E}[f_2(\mathcal{P})]\zeta_2\Big) \tag{25}$$

where $\zeta_1 = \mathrm{Var}(f(x_1, X_2) \mid x_1))$, $\zeta_2 = \mathrm{Var}(f(X_1, X_2)$, and $f_1(\mathcal{P}), f_2(\mathcal{P})$ are the number of members of $\mathcal{P} \times \mathcal{P}$ which have exactly 1 (respectively 2) indices in common.

We first consider $\mathbb{E}[f_2(\mathcal{P})]$. Recall that $\mathcal{P}$ is constructed from P permutations $\sigma_1, \ldots, \sigma_P$ of the set $\{1, \ldots, n\}$. As each index appears exactly once in each permutation, it suffices to consider the self pairs within permutations and the overlaps across pairs of permutations we get:

$$\mathbb{E}[f_2(\mathcal{P})] = \sum_{i<j}\sum_{p=1}^{P} \mathbb{E}\Big[q_{ij}^p(\mathcal{P})\big(1 + \sum_{p'\neq p} q_{ij}^{p'}(\mathcal{P})\big)\Big],$$

where $q_{ij}^p(\mathcal{P})$ is the number of pairs from the permutation p that contain $\{i, j\}$. The probability of a pair (i, j) to appear in a given permutation is $1/(n-1)$, hence using the independence between permutations we obtain:

$$\mathbb{E}[f_2(\mathcal{P})] = \frac{n(n-1)}{2}\frac{P}{n-1}\Big(1 + \frac{P-1}{n-1}\Big) = \frac{Pn}{2}\Big(1 + \frac{P-1}{n-1}\Big).$$

For $\mathbb{E}[f_1(\mathcal{P})]$, using a similar reasoning we only have to consider overlaps across each pair of permutations, in which each index pair shares exactly one index with a single pair of another permutation, except when the pair appears twice, hence:

$$\begin{aligned}
\mathbb{E}[f_1(\mathcal{P})] &= \sum_{(i,j)\in\mathcal{P}} 2(P-1) - 2\sum_{i<j}\sum_{p=1}^{P}\sum_{p'\neq p} \mathbb{E}\big[q_{ij}^p(\mathcal{P})q_{ij}^{p'}(\mathcal{P})\big], \\
&= P(P-1)n - n(n-1)\frac{P(P-1)}{(n-1)^2} \\
&= P(P-1)n\Big(1 - \frac{1}{n-1}\Big)
\end{aligned}$$

Putting everything together into (25) we get the desired result. ■

Optimal value of P. The optimal value of P depends on the kernel function, the data distribution and the privacy budget. Roughly speaking, setting P larger than 1 can be beneficial when ζ_2 is large compared to $1/\epsilon^2$. On the other hand, when $\zeta_2 = 2\zeta_1$ (which is the minimum value of ζ_2, corresponding to the extreme case where the kernel can in fact be rewritten as a sum of univariate functions (Blom, 1976)), $\mathrm{Var}(\widehat{U}_{f,n})$ simplifies to $\frac{4\zeta_1}{n} + \frac{2P}{n\epsilon^2}$ and $P = 1$ is optimal. In practice and as illustrated in our experiments, P should be set to a small constant.

Optimality of subsampling schemes. The proposed subsampling strategy is simple to implement and leads to an optimal variance, up to an additive term of $\frac{2}{Pn}\frac{P-1}{n-1}(\zeta_2 - 2\zeta_1) \geq 0$, among unbiased approximations based on $Pn/2$ pairs. Note that this additive term is 0 when $P = 1$ or $\zeta_2 = 2\zeta_1$, and is in general negligible compared to the dominating terms for small enough P. Optimal variance could be achieved at the cost of a more involved sampling scheme.[3] Alternatively, sampling schemes that can be run independently by each user without global coordination (such as sampling $P/2$ other users uniformly at random) lead to a slight increase in variance as users are not guaranteed to appear evenly across the sampled pairs.

C.2 Implementing 2PC

MPC is a subfield of cryptography concerned with the general problem of computing on private distributed data in a way in which only the result of the computation is revealed to the parties, and nothing else. In this paper the number of parties is limited to 2, and the function to be computed is $\widetilde{f}(x, y)$. There are several protocols that allow to achieve this goal, with different trade-offs in terms of security, round complexity, and also differing on how the functionality $\widetilde{f}$ is represented. These alternatives include Yao's garbled circuits (Yao, 1986; Lindell and Pinkas, 2009), the GMW protocol (Goldreich et al., 1987), and the SPDZ protocol (Damgård et al., 2011), among others. As some of the functions $\widetilde{f}$ we are interested in involve comparisons (e.g., Gini mean difference and AUC), a Boolean representation is more suitable, as it will lead to a smaller circuit. Moreover, a constant round protocol is preferred in our setting, as as users might have limited connectivity. For this reason we choose garbled circuits as our protocol, for which (Evans et al., 2018) give a detailed description including crucial practical optimizations. Moreover, we assume semi-honest adversaries in the sequel (see Goldreich, 2004, for a definition of this threat model).

Circuits for kernels. We illustrate the main ideas on Gini mean difference and AUC. As circuits for floating point arithmetic are large, they are usually avoided in MPC, to instead rely on fixed point encodings. Hence, we assume that the parties have agreed on a precision, and hence x, y are integers encoded in two's complement.

For Gini mean difference we need our 2PC protocol to compute $\texttt{fgini}(x, y) := |x - y|$. Let z be $x - y$, let $z_{k-1}, \ldots, z_0$ be the binary encoding of z, where the bitwidth k will be a constant such as 32 or 64 in practice, and let $s = z_{k-1} \cdots z_{k-1}$ be the sign bit of z replicated k times. Then $\texttt{fgini}(x, y)$ can be computed as $(z + s) \oplus s$ and, thanks to the free-XOR

[3] In addition to having each data point appear the same number of times in $\mathcal{P}$, one must ensure that no pair appears more than once.

optimization of garbled circuits (see Evans et al., 2018), the garble circuit evaluation requires only a subtraction and a summation, and thus is very efficient.

For AUC we need our 2PC protocol to compute $\texttt{fauc}(x, y) := x < y$, which requires a single comparison and thus a small number of binary gates to be evaluated in a garbled circuit.

Circuits for local randomizers. The above circuits need to be extended with output perturbation corresponding to the Laplace and randomized response mechanisms discussed above. An important observation when designing efficient circuits for these tasks is the well-known fact that a random bit with bias $1/p$, for any integer p, can be generated from only two uniform random bits suffice, in expectation. Generating a uniformly random bit is easy (and extremely cheap using garbled circuits) in the semi-honest model: each party simply generates a random bit, and then inside the circuit a random bit is reconstructed as the XOR of two bits. As XORs are for free in garbled circuits this computation is very efficient. The problem of implementing differentially private mechanisms in MPC was discussed by Dwork et al. (2006), where the authors present small circuits for sampling from an exponential distribution requiring only a $log(k)$ biased random bits, which can be constructed in parallel. Recently, Champion et al. (2019) proposed optimized constructions for several well-known differentially private mechanisms (including the geometric and Laplace mechanisms), and empirically showed their concrete efficiency.