Near-Optimal Private Approximation Protocols via a Black Box Transformation

David P. Woodruff
IBM Research-Almaden
dpwoodru@us.ibm.com

ABSTRACT

We show the following transformation: any two-party protocol for outputting a $(1+\varepsilon)$-approximation to $f(x,y) = \sum_{j=1}^{n} g(x_j, y_j)$ with probability at least $2/3$, for any non-negative efficienty computable function g, can be transformed into a two-party private approximation protocol with only a polylogarithmic factor loss in communication, computation, and round complexity. In general it is insufficient to use secure function evaluation or fully homomorphic encryption on a standard, non-private protocol for approximating f. This is because the approximation may reveal information about x and y that does not follow from $f(x,y)$. Applying our transformation and variations of it, we obtain near-optimal private approximation protocols for a wide range of problems in the data stream literature for which previously nothing was known. We give near-optimal private approximation protocols for the ℓ_p-distance for every $p \geq 0$, for the heavy hitters and importance sampling problems with respect to any ℓ_p-norm, for the max-dominance and other dominant ℓ_p-norms, for the distinct summation problem, for entropy, for cascaded frequency moments, for subspace approximation and block sampling, and for measuring independence of datasets. Using a result for data streams, we obtain private approximation protocols with polylogarithmic communication for every non-decreasing and symmetric function $g(x_j, y_j) = h(x_j - y_j)$ with at most quadratic growth. If the original (non-private) protocol is a simultaneous protocol, e.g., a sketching algorithm, then our only cryptographic assumption is efficient symmetric computationally-private information retrieval; otherwise it is fully homomorphic encryption. For all but one of these problems, the original protocol is a sketching algorithm. Our protocols generalize straightforwardly to more than two parties.

Categories and Subject Descriptors: F.1.2 Theory of Computation: Interactive and reactive computation

General Terms: Algorithms, Security, Theory

Keywords: approximation algorithms, communication complexity, cryptography, data stream algorithms

STOC'11, June 6–8, 2011, San Jose, California, USA.

1. INTRODUCTION

The availability of distributed massive datasets has led to significant privacy concerns. The use of cryptographic techniques to control access and prevent misuse of the data is needed. While generic techniques such as secure function evaluation (SFE) and fully homomorphic encryption (FHE) are available, such techniques concern exact computation, while for large datasets, computing even basic statistics exactly is prohibitive or impossible. Hence, there is a need for private approximation protocols.

Feigenbaum et al. [26] introduced the notion of a two-party private approximation protocol. Roughly speaking, a two-party protocol for a function $f(x,y)$, where the first party has input x and the second input y, is a private approximation protocol of $f(x,y)$ if it satisfies the following two properties. First, the output $F(x,y)$ must be a functionally private approximation, that is, it approximates $f(x,y)$ in the usual sense, e.g., is an (ε,δ)-approximation[1], and its distribution can be simulated given only the exact function value $f(x,y)$. Thus, a functionally private approximation captures the intuition that each party learns nothing about the other party's input from the output except what follows from $f(x,y)$ and his/her own input. The second condition of a private approximation protocol is that the entire view of the parties can be simulated given only $f(x,y)$.

In general, it is insufficient to perform secure function evaluation or fully homomorphic encryption on a standard, non-private protocol for approximating f. This is because the approximation $F(x,y)$ may reveal information about x and y that does not follow from $f(x,y)$, e.g., if $f(x,y)$ is the Hamming distance between x and y, the least significant bit of the approximation may equal an arbitrary bit of x. Given a protocol that outputs a functionally private approximation, it can be compiled in a generic way using a fully homomorphic encryption scheme of Gentry [30] to obtain a private approximation protocol by increasing the computation, communication, and round complexity by an $O^*(1)$ factor [2]. Thus, the main focus of work on private approximation protocols is on designing functionally private approximations. A functionally private approximation is also independently motivated, for instance, if two honest parties wish to publish a statistic of their joint data that is functionally private.

Similarity estimation is a basic primitive for comparing

[1] $F(x,y)$ is an (ε,δ)-approximation of $f(x,y)$ if $\forall, x, y$, $\Pr[(1-\varepsilon)f(x,y) \leq F(x,y) \leq (1+\varepsilon)f(x,y)] \geq 1-\delta$.
[2] $O^*(f)$ denotes $f(k,n,M,\varepsilon)\text{poly}(k\varepsilon^{-1}\log(nM)\log 1/\delta)$, where k is a security parameter and the $x_i, y_i \in \{-M, -M+1, \ldots, M\}$ for all $i \in [n] = \{1, 2, \ldots, n\}$.

massive data sets. A generic similarity measure between vectors $x, y \in \{-M, -M+1, \ldots, M\}^n$ is $\sum_{j=1}^n g(x_j, y_j)$, for some function g. One of the well-studied similarity measures is the ℓ_p-distance $\|x-y\|_p$ for $p \geq 0$, or equivalently, the p-th power of the ℓ_p-distance, known as the p-th frequency moment. Here the function $g(z) = |z|^p$, so that $\|x-y\|_p^p = \sum_{j=1}^n |x_j - y_j|^p$. We note that when $p = 0$, then 0^0 is interpreted as 0, and so ℓ_0 measures the number of coordinates for which x and y differ.

Various authors study private approximation protocols for the ℓ_p-distances. Feigenbaum et al. give an $O^*(\sqrt{n})$ communication protocol for privately approximating the Hamming distance between bitstrings. This was improved by Indyk and Woodruff [39] to $O^*(1)$ communication and $O^*(n^2)$ work for the Euclidean distance, for which Hamming distance on bitstrings is a special case. The work was reduced to $O^*(n)$ by Kilian et al. [44] using the fast Fourier transform. They also gave private approximation protocols for the problem of finding the ℓ_2-heavy hitters of $x-y$, and to a weaker extent the ℓ_1-heavy hitters. The latter problems are used to detect all coordinates i for which $|x_i - y_i|$ is large, see, e.g., [16, 20]. Madeira and Muthukrishnan [47] give a functionally private approximation of the ℓ_p-distance which critically relies on p-stable distributions for $p \in (0, 2]$. Nothing was known for $p \in \{0\} \cup (2, \infty)$, despite these being well-studied distances. The case $p = 0$ is known as the Hamming norm, a generalization of Hamming distance to non-binary strings, see, e.g., [18], while $p = 3$ is the skewness and $p = 4$ the kurtosis (see [1, 4, 9, 38] and papers citing/cited by these).

There are only a few other upper bounds on private approximations that we are aware of. Ishai et al. [40] introduce the multi-party model for private approximation protocols, and study a relaxed notion of privacy in it. If there are s parties with respective inputs $x^1, \ldots, x^s$, they give the simulator the aggregate vector $y = \sum_{j=1}^s x^j$ or $y = \wedge_{j=1}^s x^j$, where $\wedge$ denotes coordinate-wise minimum, rather than just giving the simulator $f(y)$. They also show the above private approximation protocols for Euclidean distance hold in the multi-party setting. Other work includes a private approximation protocol for matrix permanent and some #P-complete problems that have approximation schemes based on Monte Carlo Markov chain methods [26, 47]. For some NP-hard functions there are private approximation protocols that are not completely private, but may leak a small number of bits [34]. However, many natural NP-hard problems do not admit efficient private approximation protocols [34], even if leaking many bits [7]. The situation is even bleaker for private approximation protocols for search problems, in which the answer is not necessarily unique [6].

Thus, prior to our work, very little was known, especially for the important class of functions that admit polynomial time exact algorithms, but have much more efficient approximation algorithms, such as the large body of problems that are extensively studied in the data streaming literature; for a survey on streaming, see, e.g., [49]. This was one of the original motivations of the paper [26] which introduced private approximations, stating that "even functions that are efficiently computable for moderately sized data sets are often not efficiently computable for massive data sets." This class of functions covers practical problems in compressed sensing, information theory, numerical linear algebra, optimization, similarity estimation, and statistics.

The goal of our work is to design private approximation protocols for these functions.

1.1 Our Main Transformation

Our main result is a transformation from any two-party protocol for approximating a function $f(x, y)$ of the form $f(x, y) = \sum_{j=1}^n g(x_j, y_j)$, for any non-negative efficiently computable function g, into a private approximation protocol for $f(x, y)$ with the same communication, computation, and round complexity, up to an $O^*(1)$ factor[3]. Despite the intuition that designing private approximation protocols is more difficult than feeding a non-private approximation protocol into secure function evaluation or a fully homomorphic encryption scheme, our transformation shows there is still a generic transformation of an approximation protocol into a private one for a very large class of functions.

We first describe our transformation of a protocol into a functionally private approximation. This directly implies a private approximation protocol assuming a fully homomorphic encryption scheme. We then discuss cryptographic assumptions weaker than fully homomorphic encryption that, in the important case that the original protocol is a simultaneous protocol with shared randomness, e.g., a sketching algorithm from the data stream literature, allow us to transform our functionally private approximation into a private approximation protocol. All of our private approximation protocols have a straightforward extension to more than two parties, and only require giving the simulator the exact function value, rather than the aggregate vector. We focus on the two-party setting only for the sake of exposition and to compare with previous work. We also stress that we work in the standard cryptographic model in which the two parties do not share any randomness before the protocol begins.

The main idea is to use an efficient non-private approximation protocol for f as a black box to sample from the set $[n]$ of indices, where each $i \in [n]$ is sampled with probability within a constant factor of $\frac{g(x_i,y_i)}{f(x,y)}$. This is done by making several recursive calls to the non-private approximation protocol, which are organized in a complete binary tree with the leaves equal to the indices $i \in [n]$. Upon sampling an index $i \in [n]$, we then correct the resulting distribution by exact computation of $g(x_i, y_i)$, which can be done efficiently by exchanging the single coordinate values x_i and y_i, and computing $g(x_i, y_i)$. To correct the distribution, we design a rejection sampling procedure, which effectively adjusts our probability of sampling i to $\frac{g(x_i,y_i)}{B}$, where $B = (Mn)^{O(1)}$ is a public upper bound on $f(x, y)$. The resulting distribution is now very close (within statistical distance $\exp(-k)$, not just constant distance) to the distribution π, where π is the distribution on $[n] \cup \perp$ for which $\pi(i) = \frac{g(x_i,y_i)}{B}$ for all $i \in [n]$, and $\pi(\perp) = 1 - \sum_{i=1}^n \pi(i) = \frac{B - f(x,y)}{B}$. Here $\perp$ is a symbol indicating the sample we obtained was later rejected.

Given our sampling procedure, which we sometimes refer to as importance sampling with respect to g, we can leverage a technique for truncating information given in [39], though our application of it is simpler. We set a coin to 1 iff $\perp$ is not returned by our procedure. The coin has expectation $\frac{\sum_{j=1}^n g(x_j,y_j)}{B} = \frac{f(x,y)}{B}$. Since the distribution of a Bernoulli

[3]The computation also increases by a additive $O^*(n)$, but this does not affect the asymptotic complexity of any problem considered in this paper, since all problems here require at least linear time.

random variable is entirely determined by its expectation, these coin tosses are simulatable. We do this independently for $O^*(1)$ coins. If most of the coins are 0, then we halve B and repeat. This process of halving B depends only on the value $f(x,y)$, so is simulatable. When B is close to $f(x,y)$, with overwhelming probability a large fraction of coins will be 1, and we can (ε,δ)-approximate $f(x,y)$.

Transforming this functionally private approximation into a private approximation protocol can, as discussed above, be done using fully homomorphic encryption. However, in the case that ALG is a simultaneous protocol with shared randomness, we design a protocol under the weaker assumption of symmetric computationally-private information retrieval (SPIR) with $O^*(1)$ communication and $O^*(n)$ work. This assumption holds for almost all of our applications, which have sketching algorithms from the data stream literature. In an SPIR protocol, there is a user with an index $i \in [n] \stackrel{\text{def}}{=} \{1,2,\ldots,n\}$ and a server with a string $x \in \{0,1\}^n$ who execute a protocol for which the user learns only x_i, while a server learns nothing about i, assuming both parties must run in $\text{poly}(n)$ time. A known construction of Cachin, Micali, and Stadler [11] or of Gentry and Ramzan [31], coupled with a symmetric version due to Naor and Pinkas [51], satisfies this under the well-studied Φ-Hiding Assumption (see work citing [11]). If one is willing to lose a factor of n^γ for arbitrarily small constant γ, one can just assume additively homomorphic encryption, for which there are too many schemes to cite here.

Applications. A summary of the many results we achieve with our transformation and variations is given in Figure 1 and the next section. In that figure CC-non-private(f) denotes the non-private $O^*(1)$-round randomized communication complexity of $(O(1/\log n), 1/3)$-approximating f. We obtain optimal private approximation protocols, up to $O^*(1)$ factors, for ℓ_p-distances, ℓ_p-heavy hitters, and ℓ_p-sampling for any $p \geq 0$, entropy, max-dominance and other dominant ℓ_p-norms, distinct summation, cascaded moments, subspace approximation, block sampling, and measuring ℓ_2-independence of datasets. Except for subspace approximation and block sampling, our only assumption is SPIR with $O^*(1)$ communication and $O^*(n)$ computation. For subspace approximation and block sampling, we use fully homomorphic encryption. The same bounds hold in the multi-party setting for any $O^*(1)$ parties.

While some of these applications follow immediately from our transformation, other ones such as entropy, cascaded moments, subspace approximation, and ℓ_2 distance to independence require a few twists to our transformation.

1.2 Related Work

To better understand our work, we compare it with works on private approximations and also on differential privacy.

Indyk/Woodruff [39]. This work achieves private approximation protocols for the Hamming and Euclidean distance. It cannot handle more general functions because of the lack of a sampling procedure. Also, our method of truncating information is simpler since there it was necessary to ensure the values $g(x_i,y_i)$, for different i, are the same, up to a logarithmic factor. This was tied to their use of uniform sampling, i.e., sampling independent of the function g. Luckily for ℓ_2-distance there is a way of randomizing the instance using a random rotation so that uniform sampling is possible. In general this is not possible.

Madeira/Muthukrishnan [47]. This work extends [39] to non-negative functions f that have an efficient negligibly biased estimator that is also sharply concentrated. This means that there exists an efficient protocol ALG so that for any inputs (x,y), the output $\text{ALG}(x,y)$ is an unbiased estimator (up to $(1\pm\exp(-k))$) of $f(x,y)$, and with probability $1-\exp(-k)$, $\text{ALG}(x,y) = \tilde{\Theta}(f(x,y))$. Fortunately, the ℓ_p-norms for $p \in (0,2]$ have negligibly biased estimators that are sharply concentrated, using Li's estimator [46]. This class of functions is quite limited though. Indeed, for the functions considered in this paper it is not known how to efficiently compute sharply concentrated negligibly biased estimators. We note that the usual method of taking the median of independent repetitions does not preserve the property of being a negligibly biased estimator, and so one does not obtain a sharply concentrated negligibly biased estimator this way. In contrast, our sampling procedure works for any, neither unbiased nor concentrated, efficient protocol ALG which is capable of providing an $(O(1/\log n), 1/3)$-approximation.

Differential Privacy. An orthogonal line of research on privacy is differential privacy, which tries to capture the notion of individual privacy, see, e.g., the surveys [24, 25]. The privacy guarantees of functional and differential privacy are incomparable, and the choice between the right notion of privacy should depend on the semantics of the problem. Functional privacy asserts that if $f(z) = f(z')$ for $z \neq z'$, then the approximation of f should have the same distribution on z and z'. In contrast, the differential privacy guarantee deteriorates exponentially with the distance of z and z' under some appropriate measure of distance, usually Hamming distance. But for close z and z' with $f(z) \neq f(z')$, functional privacy gives no guarantees about the closeness of distributions of approximations of $f(z)$ and $f(z')$, while differential privacy would imply these distributions are close.

Summary of Acronyms: In the remainder of the paper, we use PAP for private approximation protocol, FPA for functionally private approximation, FHE for fully homomorphic encryption, SFE for secure function evaluation, SPIR for symmetric computationally-private information retrieval, and NBE for negligibly biased estimator.

Roadmap: In the next section we describe the proof of our main transformation. In Section 4 we give applications.

2. PRIVACY DEFINITIONS AND TOOLS

To achieve our strongest results, we need to set the security parameter $k = \text{polylog}(n)$. Thus, in the following definitions of privacy, it is insufficient to protect against $\text{poly}(k)$-time adversaries, as the parties themselves run in $\text{poly}(n)$ time. Hence, throughout we shall define security with respect to $\exp(k)$-time algorithms. We need the notion of computationaly indistinguishability.

DEFINITION 1. *Distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are computationally indistinguishable, denoted $\mathcal{D}_1 \stackrel{c}{\equiv} \mathcal{D}_2$, if for every pair of random variables $X_1 \sim \mathcal{D}_1$ and $X_2 \sim \mathcal{D}_2$ and for any family of $\exp(k)$-size circuits $\{C_k\}$, $|\Pr[C_k(X_1)=1] - \Pr[C_k(X_2)=1]| = \exp(-k)$.*

We define a two-party private protocol, as introduced in [26]; we refer the reader to [13] and [32] for more details.

We refer to the two parties as Alice and Bob. Let h be a possibly randomized mapping from input pairs (a,b)

Problem	Communication	Work
ℓ_p-distance, $p > 2$, ℓ_p-heavy hitters, and ℓ_p-sampling	$O^*(n^{1-2/p})$	$O^*(n)$
ℓ_p-distance, $p \in [0, 2]$, ℓ_p-heavy hitters, and ℓ_p-sampling	$O^*(1)$	$O^*(n)$
$\sum_{j=1}^{n} h(x_j - y_j)$, h satisfies tractability conditions of [10]	$O^*(1)$	$O^*(n)$
Max-Dominance norm and other dominant ℓ_p-norms	$O^*(1)$	$O^*(n)$
Distinct Summation	$O^*(1)$	$O^*(n)$
Entropy	$O^*(1)$	$O^*(n)$
Cascaded Moments $F_q(F_p)$ of $n \times d$ matrix, every q and p	$O^*(\text{CC-non-private}(F_q(F_p)))$	$O^*(nd)$
Subspace Approximation of n points in $\mathbb{R}^d$ and Block Sampling	$O^*(d)$	$O^*(nd)$
ℓ_2-Distance to Independence	$O^*(1)$	$O^*(n^2)$

Figure 1: For each we obtain optimal (up to $O^*(1)$ factors) relative error PAPs. All are $O^*(1)$ rounds.

to output pairs (c, d). A randomized synchronous protocol proceeds in rounds. In each round a party sends a message based on the security parameter k, his/her input and his/her random tape, as well as messages passed in previous rounds. During each round either party may decide to terminate based on his/her view, where here a party's view is its input, random tape, and all messages exchanged.

For a protocol Π for a mapping h, let $\mathsf{REAL}_{\Pi,A}(k,(a,b))$ be a random variable which contains the view of Alice in Π when the input to the protocol is (a, b), concatenated with the output of Bob (this concatenation is required for technical reasons). We similarly define $\mathsf{REAL}_{\Pi,B}(k,(a,b))$. Next, for an efficient (poly(n)-time) algorithm S known as a simulator, let $\mathsf{IDEAL}_{\Pi,A,S,h}(k,(a,b))$ be the output of the random process: (1) apply h to (a, b), resulting in a pair of outputs (c, d), (2) invoke S on (k, a, c), (3) concatenate the output of S with d. Similarly define $\mathsf{IDEAL}_{\Pi,B,S,h}(k,(a,b))$.

DEFINITION 2. *A private two-party protocol Π of a randomized mapping h is a protocol for which: (1) the distribution on outputs has ℓ_1-distance* $\exp(-k)$ *from that of h, and (2) there is a* poly(n)*-time simulator S_A such that for any input pair (a, b), we have $\{\mathsf{REAL}_{\Pi,A}(k,(a,b))\}_{k\in\mathbb{N}} \stackrel{c}{\equiv} \{\mathsf{IDEAL}_{\Pi,A,S_A,h}(k,(a,b))\}_{k\in\mathbb{N}}$. There must also be an efficient simulator S_B with the analogous property for Bob.*

We now define an SPIR protocol. Alice has a string $a \in \{0,1\}^n$ while Bob has an index $i \in [n]$. The randomized mapping is $h(a, i) = a_i$, and an SPIR protocol is a private protocol for h. It is known how to construct an SPIR protocol from a PIR protocol, namely, a protocol for SPIR which relaxes privacy to only require that there is a simulator S_B in definition 2, rather than both simulators S_A and S_B. The PIR to SPIR transformation only incurs an $O^*(1)$ factor blowup in communication, computation, and number of rounds; see the work by Naor and Pinkas [51]. Let $C(n)$ be the communication of a PIR protocol with $n \cdot \text{polylog}(n)$ work per party and polylog(n) rounds. $C(n)$ can be as low as polylog(n), see, e.g., [11]. We assume such a scheme exists in the remainder of the paper. We also need a definition and a theorem of Naor and Nissim [50].

DEFINITION 3. *([50]) Two parties are said to jointly evaluate a circuit with ROM if the randomized mapping h the parties compute can be implemented as a circuit with at most* poly(n, k) *gates of the following types. The gates can be either NAND gates (or gates defining any complete basis on bits), or so-called lookup gates. To define a lookup gate, Alice (resp. Bob) builds a table $R_A \in \{0,1\}^n$ (resp. R_B), and the lookup gate, given a pair (A, j) (resp. (B, j)), outputs $R_A(j)$ (resp. $R_B(j)$).*

THEOREM 4. *([50]) Given a PIR (and hence an SPIR) scheme with $C(n) =$* polylog(n)*, any circuit with ROM Λ can be privately computed with $|\Lambda| \cdot$* polylog(n) *communication, $n \cdot |\Lambda| \cdot$* polylog(n) *work, and $|\Lambda| \cdot$* polylog(n) *rounds, where $|\Lambda|$ is the # of gates in Λ.*

Finally, we use a standard composition theorem [13, 32].

DEFINITION 5. *An oracle-aided protocol using an oracle functionality $\mathcal{O}$ (i.e., the parties have an oracle tape for which they can provide inputs and receive output from the oracle) privately computes h if there are simulators S_A, S_B as in Definition 2, where the corresponding views of the parties are defined in the natural manner to include oracle answers.*

THEOREM 6. *([13, 32]) Suppose there is a private oracle-aided protocol for h given oracle functionality $\mathcal{O}$, and a private protocol for computing h. Then the protocol defined by replacing each oracle-call to $\mathcal{O}$ by a protocol that privately computes $\mathcal{O}$ is a private protocol for h.*

Two parties output a *secret-sharing* [32] of a function f with output pair (c, d), where c and d are bitstrings of length s_c and s_d, respectively, if Alice's output is a pair of random strings $r_A(c) \in \{0,1\}^{s_c}$ and $r_A(d) \in \{0,1\}^{s_d}$, while Bob's output is a pair of random strings $r_B(c) \in \{0,1\}^{s_c}$ and $r_B(d) \in \{0,1\}^{s_d}$, with the property that $r_A(c) \oplus r_B(c) = c$ and $r_A(d) \oplus r_B(d) = d$, but otherwise the strings $r_A(c)$, $r_A(d)$, $r_B(c)$, and $r_B(d)$ are random.

A non-private protocol is a *simultaneous protocol* [45] for (ε, δ)-approximation of a function $h(a, b)$ if Alice and Bob share public randomness, Alice sends a single message M_A to a referee, Bob sends a single message M_B to the referee, and the referee with the public randomness, M_A, and M_B, generates an (ε, δ)-approximation to $h(a, b)$. If, in addition, a and b are n-dimensional vectors and there is a distribution over matrices W with n columns, chosen independently of a, b, for which $M_A = W \cdot a$ and $M_B = W \cdot b$, then we call the protocol a *sketching algorithm* [49]. We stress that all of our PAPs will be designed in the standard model, i.e., with no shared public randomness, even if the original non-private protocol uses shared randomness.

3. MAIN TRANSFORMATION

In our protocols, the parties must run in poly$(nk\varepsilon^{-1} \log M)$ time. We can assume that $\varepsilon > 1/\text{poly}(n)$, as otherwise it

would become more efficient to compute $\sum_{j=1}^{n} g(x_j, y_j)$ exactly using known secure function evaluation techniques [32]. The security parameter k will be polylog(n) or n^{γ} for arbitrarily small constant $\gamma > 0$. For simplicity we also assume $\log M \leq \text{poly}(n)$. It follows that the parties must run in poly(n) time. We can, w.l.o.g., assume that both parties are semi-honest, meaning they follow the protocol but may keep message histories in an attempt to learn more than what is prescribed. In Section 6 of [50], it is shown how to transform a semi-honest protocol into a protocol secure in the malicious model, at the cost of at most an $O^*(1)$ factor.

Feigenbaum et al. [26] define the following.

DEFINITION 7. *A function h' is functionally private with respect to a function h if there is a* poly(n)*-time simulator S for which for any input x, $\{S(h(x))\} \stackrel{c}{\equiv} \{h'(x)\}$.*

DEFINITION 8. *A two-party private (ε, δ)-approximation protocol of h is a private protocol (see Definition 2) that computes a randomized mapping $\hat{h}$ satisfying the following two properties: 1. $\hat{h}$ is functionally private for h, and 2. $\hat{h}$ is an (ε, δ)-approximation of h.*

We assume, w.l.o.g., that n is a power of 2. We start by formally defining importance sampling w.r.t. g.

DEFINITION 9. *In the g-sampling functionality, both parties receive integers B and k. Alice receives an input $x \in \{-M, -M+1, \ldots M\}^n$, while Bob receives an input $y \in \{-M, -M+1, \ldots, M\}^n$. There is a promise that $B \geq 2\sum_{j=1}^{n} g(x_j, y_j) = 2f(x, y)$. Define the distribution π on $[n] \cup \perp$, where $\pi(i) = \frac{g(x_i, y_i)}{B}$ for all $i \in [n]$, and $\pi(\perp) = 1 - \sum_{i=1}^{n} \pi(i) = \frac{B - f(x,y)}{B}$. The output is a secret-sharing of a random $I \in [n] \cup \{\perp\}$ from a distribution π' with $\|\pi' - \pi\|_1 \leq \exp(-k)$.*

Let Alg$(n', \varepsilon', \delta')$ be a protocol for (ε', δ')-approximating $\sum_j g(x_j, y_j)$ on n' coordinates. Suppose Alg has $r(n', \varepsilon', \delta')$ rounds, $c(n', \varepsilon', \delta')$ communication, and $t(n', \varepsilon', \delta')$ time. We show g-SAMPLER in Figure 2 privately implements g-sampling.

LEMMA 10. *Protocol g-SAMPLER correctly implements the g-sampling functionality.*

PROOF. Let I be the value secret-shared by the two parties upon termination of the protocol, assuming it does not output fail. We need to show that I is sampled from a distribution π' that has ℓ_1 distance $\exp(-k)$ from π. Consider the complete binary tree $\mathcal{T}$ on coordinate set $[n]$, and consider the $2n-1$ subsets S_v associated with nodes v of $\mathcal{T}$. Since $\delta = \exp(-k)$, by a union bound, for any subset S_v of coordinates associated with a node v of $\mathcal{T}$, ALG on vectors x, y restricted to coordinates in S_v succeeds in providing a $(1 \pm \zeta)$-approximation with probability at least $1 - (2n-1)\exp(-k) = 1 - \exp(-k)$. Fix the random string σ used by the protocol, and condition on the event $\mathcal{E}$ of it having this property. The protocol does not invoke Alg on all subsets S_v, though we assume it is correct on all such S_v.

Fixing σ, all invocations of Alg become deterministic, and so for each node $v \in \mathcal{T}$, there is a well-defined probability r_v, over the coin tosses of the binary search in step 2(c)iv, that the protocol reaches node v. Namely, suppose v is at shortest path distance ℓ from the root v_0 of $\mathcal{T}$. Let $v_0, v_1, v_2, \ldots, v_\ell = v$ be the unique path from the root of $\mathcal{T}$ to v. Let $w_1, w_2, \ldots, \ldots, w_\ell$ be the siblings of $v_1, v_2, \ldots, v_\ell$, respectively. Then, $r_v = \prod_{i=1}^{\ell} \frac{p_{v_i}}{p_{v_i} + p_{w_i}}$, where the p_{v_i} are as defined in step 2(c)iii. Note if the denominator is 0, then the numerator is also 0, and in this case the probability is 0.

Since we condition on event $\mathcal{E}$, using the non-negativity of g, we obtain a telescoping product:

$$\begin{aligned} r_v &= \prod_{i=1}^{\ell} \frac{p_{v_i}}{p_{v_i} + p_{w_i}} \\ &\leq \frac{(1+\zeta)^{\ell}}{(1-\zeta)^{\ell}} \prod_{i=1}^{\ell} \frac{\sum_{j \in S_{v_i}} g(x_j, y_j)}{\sum_{j \in S_{v_i}} g(x_j, y_j) + \sum_{j \in S_{w_i}} g(x_j, y_j)} \\ &= \frac{(1+\zeta)^{\ell}}{(1-\zeta)^{\ell}} \cdot \frac{\sum_{j \in S_v} g(x_j, y_j)}{\sum_{j=1}^{n} g(x_j, y_j)} \leq 2 \cdot \frac{\sum_{j \in S_v} g(x_j, y_j)}{\sum_{j=1}^{n} g(x_j, y_j)}, \end{aligned}$$

for a small enough $\zeta = \Theta(1/\log n)$. An analogous argument shows also that $r_v \geq \frac{1}{2} \cdot \frac{\sum_{j \in S_v} g(x_j, y_j)}{\sum_{j=1}^{n} g(x_j, y_j)}$. Notice that these bounds on r_v also hold if $\sum_{j \in S_v} g(x_j, y_j) = 0$. Now, in step 4(c), we are promised that $B \geq 2\sum_{j=1}^{n} g(x_j, y_j)$, so $p \leq \frac{g(x_q, y_q)}{2\beta \sum_{j=1}^{n} g(x_j, y_j)}$. But $\beta = r_q$ for a leaf $q \in \mathcal{T}$, and by the above $r_q \geq \frac{1}{2} \cdot \frac{g(x_q, y_q)}{\sum_{j=1}^{n} g(x_j, y_j)}$, and so $p \leq 1$. Hence, we do not output fail in step 4(c). It follows, for our fixed choice of σ, that the probability we output coordinate $I = i$ is $r_i \cdot \frac{g(x_i, y_i)}{B r_i} = \frac{g(x_i, y_j)}{B}$. Since we have a distribution, for fixed σ, it follows that $\Pr[I = \perp] = 1 - \frac{\sum_{j=1}^{n} g(x_j, y_j)}{B}$. Event $\mathcal{E}$ occurs with probability $1 - \exp(-k)$, and the above holds for any choice of σ for which $\mathcal{E}$ occurs. □

We prove the following lemmas. Note that the first needs to be shown even with γ in the parties' views.

LEMMA 11. *Protocol g-SAMPLER privately implements the g-sampling functionality.*

PROOF. We first argue the case when ALG is a simultaneous protocol. In this case, a party's view consists of the seed γ of the generator G, and a collection of secret shares output from the secure circuit with ROM evaluations. We can apply Theorem 6 a total of $\log n + 1$ times, each time using Theorem 4. The only difficulty is that $\mathsf{REAL}_{g-sampler,A}(k, (a, b))$ contains the view of Alice *concatenated with the output of Bob*, and therefore we must prove the distribution of I conditioned on γ has ℓ_1-distance $\exp(-k)$ from π. This follows from Lemma 10, since if the event $\mathcal{E}$ in the proof of Lemma 10 occurs, then for any value of γ and hence the value $G(\gamma)$ of the pseudorandom generator, the random variable I is distributed according to π. Since $\mathcal{E}$ occurs with probability $1 - \exp(-k)$, this shows the simulator for Alice S_A can just output a random γ, in addition to the output of the simulator of Theorem 4. Hence, there is a simulator S_A as required by Definition 2.

If ALG is a general protocol, we instead implement the entire g-SAMPLER protocol using FHE. The lemma immediately follows by the properties of FHE [30]. □

LEMMA 12. *For $\zeta = \Theta(1/\log n)$, protocol g-SAMPLER can be implemented in $O^*(c(n, \zeta, 1/3))$ communication, a total of $O^*(t(n, \zeta, 1/3) + n)$ time, and $O^*(r(n, \zeta, 1/3))$ rounds.*

PROOF. We first argue this in the case that ALG is a simultaneous protocol. In this case, There are $\log n$ iterations of step 2. In the j-th iteration, both parties invoke Alg

Input: Alice is given $x \in \{-M, \ldots, M\}^n$ and 1^k, while Bob is given $y \in \{-M, \ldots, M\}^n$ and 1^k.
Both parties are given an integer $B \geq 2\sum_{j=1}^{n} g(x_j, y_j)$.
Output: The parties output a secret-sharing of a random $I \in [n] \cup \{\perp\}$ from a distribution statistically close to:
$\forall i,\ \Pr[I = i] = \frac{g(x_i, y_i)}{B}$, and $\Pr[\perp] = 1 - \sum_{j=1}^{n} \frac{g(x_j, y_j)}{B}$.

1. Initialize $S = [n]$, $\delta = \exp(-k)$, $\zeta = \Theta(\frac{1}{\log n})$, $\beta = 1$, and q to be a pointer to the root of a complete binary tree on n leaves.
   Let G be a PRG stretching $O^*(1)$ bits to $O^*(n)$ bits secure against poly(n)-sized circuits that can be evaluated in $O^*(n)$ time. Such G are implied by our assumption on SPIR, see, Remark 7 in [39].
   Alice sends Bob a seed γ to G, from which the parties share the random string $G(\gamma) = \sigma$.
2. For $j = 1, 2, \ldots, \log n$, in the j-th iteration do:
   (a) Alice and Bob break the coordinate set $[n]$ into $\frac{n}{2^j}$ contiguous blocks of coordinates $x^1, \ldots, x^{2^j}$ and $y^1, \ldots, y^{2^j}$, respectively.
   (b) Alice and Bob respectively execute $\mathrm{Alg}(\frac{n}{2^j}, \zeta, \delta)$ on x^ℓ and y^ℓ for each $\ell \in [2^j]$, using σ as the randomness for each execution. Let the resulting states of Alg be $\mathrm{state}_A(1)$, $\mathrm{state}_A(2), \ldots,$ $\mathrm{state}_A(2^j)$ and $\mathrm{state}_B(1)$, $\mathrm{state}_B(2), \ldots,$ $\mathrm{state}_B(2^j)$, which are the ROM tables of the parties.
   (c) A secure circuit with ROM performs the following computation:
      i. It maintains the state of q internally (it is secret-shared between the two parties).
      ii. Viewing the set $[2^j]$ as the internal nodes in the j-th level of a complete binary tree, it uses SPIR to retrieve $\mathrm{state}_A(L)$, $\mathrm{state}_A(R)$, $\mathrm{state}_B(L)$ and $\mathrm{state}_B(R)$, where L and R are the left and right child of q, respectively.
      iii. It combines $\mathrm{state}_A(L)$ and $\mathrm{state}_B(L)$ to obtain $p_L = \mathrm{Alg}(\frac{n}{2^j}, \zeta, \delta)(x^L, y^L)$. It combines $\mathrm{state}_A(R)$ and $\mathrm{state}_B(R)$ to obtain $p_R = \mathrm{Alg}(\frac{n}{2^j}, \zeta, \delta)(x^R, y^R)$.
      iv. Suppose first that $(p_L, p_R) \neq (0, 0)$. It sets q to point to L with probability $\frac{p_L}{p_L + p_R}$, and otherwise sets q to point to R. In the first case it sets $\beta = \beta \cdot \frac{p_L}{p_L + p_R}$. In the second case, it sets $\beta = \beta \cdot \frac{p_R}{p_L + p_R}$. If $(p_L, p_R) = (0, 0)$, it outputs a pointer q to $\perp$ and β remains the same.
      v. If $j = \log n$, it outputs a secret-sharing (e, f) of q and β to the two parties.
3. Alice and Bob create ROM tables for the entries of x and y respectively.
4. A secure circuit with ROM performs the following algorithm:
   (a) It uses inputs e and f to reconstruct q and β. If q points to $\perp$, it outputs a secret-sharing of $\perp$ to the two parties.
   (b) Otherwise, it uses SPIR to retrieve x_q and y_q, and computes $g(x_q, y_q)$.
   (c) Put $p = \frac{g(x_q, y_q)}{B \cdot \beta}$. If $p > 1$, output fail.
   (d) Otherwise, with probability p, output a secret-sharing of q to the two parties, else output a secret-sharing of $\perp$.
5. The parties output the output of the secure circuit evaluation with ROM in step 4.

Figure 2: Our protocol g-Sampler implementing the g-sampling functionality for simultaneous protocols Alg. If Alg is not a simultaneous protocol, we can instead implement the entire protocol using FHE. In the j-th iteration of step 2, Alice and Bob will only execute $\mathrm{Alg}(\frac{n}{2^j}, \zeta, \delta)$ on the left and right child, L and R, of q. By the properties of FHE, these values L and R are unknown to the parties.

Main Protocol:
Input: Alice is given $x \in \{-M, \ldots, M\}^n$ and 1^k, while Bob is given $y \in \{-M, \ldots, M\}^n$, and 1^k.
Output: A private (ε, δ)-approximation protocol for $f(x, y) = \sum_{j=1}^{n} g(x_j, y_j)$.

1. Let B be a public upper bound on $f(x, y)$, for any possible inputs x, y. We assume $\log(B) = O^*(1)$. Let $\ell = O^*(1)$ be sufficiently large.
2. Repeat the following in a secure circuit with ROM:
   (a) For $j \in [\ell]$, independently run g-sampler$(x, y, 1^k)$, let the output be shares of $I_j \in [n] \cup \{\perp\}$.
   (b) Independently generate ℓ coin tosses $z_1, \ldots, z_\ell$, where $z_j = 1$ iff $I_j \neq \perp$.
   (c) $B = B/2$.
3. Until $\sum_{j=1}^{\ell} z_j \geq \frac{\ell}{8}$ or $B < 1$.
4. Output $\Psi = \frac{2B}{\ell} \sum_{j=1}^{\ell} z_j$.

2^j times on inputs of size $n/2^j$ to achieve a $(\zeta, \exp(-k))$-approximation. Note that $c(n,\zeta,\delta) = O(k)\cdot c(n,\zeta,1/3)$, $t(n,\zeta,\delta) = O(k)\cdot t(n,\zeta,1/3)$, and $r(n,\zeta,\delta) = O(k)\cdot r(n,\zeta,1/3)$, since we may independently repeat Alg $O(\log 1/\delta)$ times and take the median of its outputs.

Step 3 and step 4 can be done in $O^*(1)$ communication, $O^*(n)$ time, and $O(1)$ rounds, given our assumption of an efficient SPIR protocol. Assuming we use an efficient SPIR protocol to retrieve each bit of the state of Alg, the communication is $O^*(1)\cdot\sum_{j=1}^{\log n} c(n2^{-j},\zeta,1/3) = O^*(c(n,\zeta,1/3))$. The number of rounds is $O^*(1)\cdot\sum_{j=1}^{\log n} r(n2^{-j},\zeta,1/3) = O^*(r(n,\zeta,1/3))$. Finally, the time is $O^*(n) + \sum_{j=1}^{\log n} 2^j \cdot t(n2^{-j},\zeta,1/3)$. If $t(n',\zeta,1/3) = \tilde{\Omega}(n')$, then this sum is $O^*(t(n,\zeta,1/3))$. Otherwise, the additive $O^*(n)$ dominates.

For the case that ALG is a general protocol, we instead implement the entire g-SAMPLER protocol using FHE. Notice that in the j-th iteration of step 2, Alice and Bob will only execute ALG$(\frac{n}{2^j},\zeta,\delta)$ on the left and right child, L and R, of q. FHE only increases communication, round, and time complexities by an $O^*(k)$ factor (assuming the original time complexity is at at least linear). □

THEOREM 13. *For $\zeta = \Theta(1/\log n)$, the protocol* MAIN *is a PAP for f, i.e., an (ε,δ)-FPA, and a private protocol with $O^*(c(n,\zeta,1/3))$ communication, $O^*(t(n,\zeta,1/3)+n)$ time, and $O^*(r(n,\zeta,1/3))$ rounds.*

PROOF. We first show that MAIN outputs an $(\varepsilon,\exp(-k))$-approximation of $\sum_{j=1}^n g(x_j,y_j)$. Observe that by Lemma 10, in any iteration and for any $j\in[\ell]$, $\mathbf{E}[Z_j] = (1\pm\exp(-k))\frac{\sum_{j=1}^n g(x_j,y_j)}{B}$. Since B is halved in step 2c, by linearity of expectation, $\mathbf{E}[\Psi] = \sum_{j=1}^n g(x_j,y_j)$. For the concentration, with probability $1-\exp(-k)$, if $B \geq \Theta(k)\cdot\sum_{j=1}^n g(x_j,y_j)$, then $\sum_{j=1}^{\ell} z_j < \frac{\ell}{8}$. On the other hand, if $B = O(k)\cdot\sum_{j=1}^n g(x_j,y_j)$, then for sufficiently large $\ell = O^*(1)$, by a Chernoff bound we have

$$\Pr\left[\left|\sum_{j=1}^{\ell} z_j - \mathbf{E}[\sum_{j=1}^{\ell} z_j]\right| > \varepsilon\mathbf{E}[\sum_{j=1}^{\ell} z_j]\right] \leq \exp(-k),$$

and by a union bound we can assume this holds for all such values of B. If $\sum_{j=1}^n g(x_j,y_j) = 0$, MAIN outputs 0. Else, there is a B for which $\mathbf{E}[\sum_{j=1}^{\ell} z_j] \geq \frac{\ell}{4}$, it follows that in step 3 we will have $\sum_{j=1}^{\ell} z_j \geq \frac{\ell}{8}$, and this sum provides a $(1\pm\varepsilon)$-approximation to $\mathbf{E}[\sum_{j=1}^{\ell} z_j] = \frac{\ell}{2B}\sum_{j=1}^n g(x_j,y_j)$ with probability $1-\exp(-k)$.

Next, we show that MAIN is functionally private. We describe the simulator S in the figure below.

The simulator S is given $f(x,y)$.

1. Let B be an upper bound on $f(x,y)$, for any possible inputs x,y. We assume $\log B = O^*(1)$. Let $\ell = O^*(1)$ be sufficiently large.
2. Repeat the following:
   (a) For $j\in[\ell]$, generate ℓ independent coin tosses z_j with bias $\frac{f(x,y)}{B}$.
   (b) $B = B/2$.
3. Until $\sum_{j=1}^{\ell} z_j \geq \frac{\ell}{8}$ or $B<1$.
4. Output $\Psi' = \frac{2B}{\ell}\sum_{j=1}^{\ell} z_j$.

Notice that the probabilities $z_j = 1$ in the simulated and the real view differ only by a factor of $1\pm\exp(-k)$. It follows that the distributions of Ψ and Ψ' have ℓ_1-distance $\exp(-k)$, which completes the proof.

We next argue that the protocol is private and efficient. We argue that MAIN satisfies the requirements of Definition 2. The first part follows from the above. By Lemma 11 and 6, we can replace the calls to g-SAMPLER with an oracle functionality. By Theorem 4, the functionality in step 2 can be implemented privately.

For the efficiency, there is only an $O^*(1)$ overhead in each of these measures from that of protocol g-SAMPLER, so the lemma follows by Lemma 12. □

4. APPLICATIONS

We say a value is *near-optimal* if it is optimal up to an $O^*(1)$ factor. We say a PAP is near-optimal if its communication, computation, and round complexity are simultaneously optimal up to an $O^*(1)$ factor. For all problems we consider, we obtain near-optimal PAPs. For brevity, we sometimes describe our PAPs as FPAs, mentioning any subtleties needed to implement the FPA as a PAP using SPIR. In the interest of space, for some applications we just give proof sketches, deferring the formal proofs to the full version.

ℓ_p-Distances. Combining our transformation with ℓ_p-estimation algorithms [9, 38], for $g(x_j,y_j) = |x_j - y_j|^p$ we obtain near-optimal $O^*(n^{1-2/p})$ communication, $O^*(n)$ computation, and $O^*(1)$ round PAPs for the ℓ_p-distance, $p>2$, as well as a near-optimal $O^*(1)$ communication, $O^*(n)$ computation, and $O^*(1)$ round PAP for the ℓ_0-distance. No sublinear communication PAPs were known for these problems, see the references above. This was the main motivation for this work. Existing algorithms for these ℓ_p and their heavy hitters (see below) are blatantly not functionally private. The difference with $p\in\{0\}\cup(2,\infty)$ is that p-stable distributions do not exist, making algorithms for them much more complicated, and making FPAs harder to design. We overcome this since our reduction is black box.

Even though PAPs or FPAs are known for $p\in(0,2]$, our framework has several advantages. One advantage is that we transform any protocol for ℓ_p into a PAP, making new tradeoffs possible. We can use protocols more suitable for inputs given as a list of ranges [5, 12, 27, 53], with faster update time [42, 52], or that use less randomness [42, 43]. For example, we improve the update time of [47] for ℓ_2 by a factor of k using the algorithm of [59] with $\varepsilon = 1/\log n$ (to do binary search), while for $p\in(0,2)$ we improve [47] by a factor of $k/\text{poly}(\log\log n)$ using the algorithm of [42]. Our communication is a factor of $\log^2 n/k$ times that of [47], since we lose a $\log^2 n$ factor since $\varepsilon = 1/\log n$ as opposed to $\varepsilon = 1/2$ in [47], but we gain a factor of k since we do not need k seeds for random functions as in [47]. Another advantage is our transformation avoids rounding issues of real numbers needed to ensure functional privacy in previous work [26, 39, 47]; in our case the parties can compute $g(x_i,y_i)$ to arbitrary precision after communicating x_i and y_i, where i is the coordinate sampled by g-SAMPLER.

Heavy Hitters and Compressed Sensing. Letting $z = x - y$, we want an r-sparse vector $\tilde{z}$ with $\|z-\tilde{z}\|_p^p \leq (1+\varepsilon)\|z - z_{opt}\|_p^p$, where z_{opt} is an r-sparse vector minimizing $\|z-z_{opt}\|_p^p$. In [44], the authors show that if only z_{opt} is leaked, then $\Omega(n)$ communication is requred. The authors relax the problem by allowing $\|z\|_2$ to also be leaked, and

show how to near-optimally solve the heavy hitters problem for $p \in \{1, 2\}$ in this case. For $p = 2$ they argue this is in fact desirable, since the leakage is equivalent to $\|z\|_2^2 - \|\tilde{z}\|_2^2 = \|\tilde{z} - z\|_2^2$, the error incurred of the r-sparse representation, which is a common thing to want (equality holds since we can assume $\tilde{z}$ agrees with z on its non-zero coordinates).

Plugging our PAPs for ℓ_p-distances into the main protocol in [44], we improve this by showing how to near-optimally solve the problem of finding $\tilde{z}$ with $\|\tilde{z} - z\|_p^p \leq (1+\varepsilon)\|z_{opt} - z\|_p^p$ leaking z_{opt} and $\|z\|_p^p$ for every $p \geq 0$. If $p \in [0, 2]$, the communication is $O^*(1)$, while if $p > 2$ the communication is $O^*(n^{1-2/p})$, which is required [4]. We note that the information we leak is more natural than that leaked in [44], who for $p = 1$ leak $\|z\|_2$ and $\tilde{z}$ rather than $\|z\|_1$ and $\tilde{z}$, the latter being equivalent to leaking $\|z - \tilde{z}\|_1$ and $\tilde{z}$, the error incurred by the sparse representation. One minor point is that we need a non-private near-optimal heavy-hitters protocol for every ℓ_p. For $p \in [0, 2]$ this is given in, e.g., Corollary 3.1 of [48]. For $p > 2$ there is an implicit algorithm with $O^*(n^{1-2/p})$ space in [38], which is optimal [4, 54].

General Similarity Measures. While our transformation gives near-optimal PAPs for any function of the form $f(x, y) = \sum_{j=1}^n g(x_j, y_j)$, for non-negative g, we may want to know for which g we obtain PAPs with $O^*(1)$ computation, $O^*(n)$ computation, and $O^*(1)$ rounds. For this we can use a theorem of Braverman and Ostrovsky [10] which very roughly says that if $g(x_j, y_j) = h(x_j - y_j) = O((x_j - y_j)^2)$ and h satisfies a few additional restrictions, then $f(x, y)$ can be computed in $O^*(1)$ space, 1-pass, and $O^*(n)$ time (assuming h can be computed in $O^*(1)$ time). Applying our transformation, we obtain PAPs with the aforementioned resources for any such h, which includes functions as bizarre as $h(x) = (x(x+1))^{.5\arctan(x+1)}$. We omit the details.

Max-Dominance Norm, Dominant ℓ_p-norms, and Distinct Summation. The Max-Dominance Norm is useful in financial applications and IP network monitoring [19]. Alice has $x \in \{0, 1, \ldots, M\}^n$, Bob has $y \in \{0, 1, \ldots, M\}^n$, and the max-dominance norm is $\sum_{j=1}^n \max(x_j, y_j)$. This problem, and its generalization, the dominant ℓ_p-norm, defined as $(\sum_{j=1}^n \max(x_j, y_j)^p)^{1/p}$ for $p > 0$, are studied in [19, 53, 56, 57, 58] (in [40] this problem is instead studied for $p < 0$, which is useful for coordinatewise minima). There are no sharply concentrated NBEs known for $p > 0$. For example, the estimators Z of [56] are distributed as p-Fréchet, which, if the dominant ℓ_p-norm is c, have $\Pr[Z > z] = 1 - \exp(-c^p z^{-p})$. For $p \leq 1$, there is no expectation, while for general p these are heavy-tailed, so there is a non-negligible $(1/\text{poly}(n))$ probability of observing a value that is $\text{poly}(n)$ times c. Nevertheless, the references above give (ε, δ)-approximations for these problems in $O^*(1)$ space, and by our transformation, we obtain near-optimal PAPs. We also get a near-optimal PAP for the related distinct summation problem in sensor networks [53], which also does not have a sharply concentrated NBE. Here, for each $j \in [n]$ there is a $v_j \in \{1, \ldots M\}$ and Alice has either (j, v_j) or $(j, 0)$, while Bob has either (j, v_j) or $(j, 0)$. The problem is to compute $\sum_{\text{distinct } (j,v_j)} v_j$, that is, for each j, either the value v_j or 0 contributes to the sum.

Entropy with Relative Error. Entropy $H(x, y) = \sum_{i=1}^n \frac{x_i+y_i}{\sum_{j=1}^n x_j+y_j} \cdot \log \frac{\sum_{j=1}^n x_j+y_j}{x_i+y_i}$ is defined for inputs x, y with $(x + y)_i \in \mathbb{R}^{\geq 0}$ for all $i \in [n]$. Here, if $x_i + y_i = 0$, we (as usual) interpret $0 \log \frac{1}{0}$ as 0. We allow x_i or y_i to be negative, but require their sum to be non-negative. This is the strict turnstile model in streaming, for which entropy is well-studied [8, 14, 15, 33, 36], and sketching algorithms with relative error, $O^*(1)$ space and update time [8, 36] are known. There are no known NBEs concentrated enough to achieve relative error. The natural NBE is to sample a coordinate i with probability $\frac{x_i+y_i}{\sum_{j=1}^n x_j+y_j}$ and output $\log \frac{\sum_{j=1}^n x_j+y_j}{x_i+y_i}$. However, while the estimator is unbiased, the concentration is poor and can only be used to achieve additive error. We will achieve relative error. $H(x, y)$ is not in the class of functions handled by our transformation. The important observation is that for any parameter $T \geq \sum_{j=1}^n x_j + y_j$, the function $H_T(x, y) = \sum_{i=1}^n \frac{x_i+y_i}{T} \cdot \log \frac{T}{x_i+y_i}$ also has an efficient relative error algorithm, given the values T and $\sum_{j=1}^n x_j + y_j$. Indeed, we run an efficient algorithm for $H(x, y)$, get $\hat{H}$, and output $\frac{\sum_{j=1}^n x_j+y_j}{T} \cdot \hat{H} + \frac{\sum_{j=1}^n x_j+y_j}{T} \cdot \log\left(\frac{T}{\sum_{j=1}^n x_j+y_j}\right)$. The additive error is at most $\varepsilon \cdot \frac{\sum_{j=1}^n x_j+y_j}{T} H(x, y) = \varepsilon \sum_{i=1}^n \frac{x_i+y_i}{T} \cdot \log \frac{\sum_{j=1}^n x_j+y_j}{x_i+y_i} \leq \varepsilon \sum_{i=1}^n \frac{x_i+y_i}{T} \cdot \log \frac{T}{x_i+y_i} = \varepsilon H_T(x, y)$. We fix $T = \sum_{j=1}^n x_j + y_j$ and in recursive calls in the binary search use the same value of T rather than $\sum_{j \in S} x_j + y_j$ for the set S under consideration (so we recursively compute H_T rather than H). In the outer level of recursion, $H(x, y) = H_T(x, y)$, and H_T has the form of our transformation, so we get a PAP for $H(x, y)$ with relative error. We do not need FHE, since we can obtain $\sum_{j=1}^n x_j + y_j$ using SFE.

Subspace Approximation and Sampling Blocks. Approximating a pointset by a subspace is studied in the linear algebra community. The form we consider is in [22, 23, 28, 29, 35, 55] in the form of approximation to a fixed subspace. For more references and connections to regression, see the references therein. In our setting Alice has $n \times d$ matrix A, Bob has $n \times d$ matrix B, and $C = A + B$, representing n records each with d attributes. They want to secret share a coreset, i.e., a small weighted subset of rows of C so that later, for any fixed j-dimensional subspace F of $\mathbb{R}^d$, $\text{cost}(C, F) = \sum_{i=1}^n \text{dist}(C_i, F)$ can be $(1 + \varepsilon)$-approximated from the coreset with functional privacy and probability $1 - \exp(-k)$. dist is ℓ_2-distance of a point to a subspace.

We first review a coreset construction of [29], the main algorithms being DimReduction and AdaptiveSampling algorithms given there. Assume the dimension j of the query subspace is constant. The authors efficiently obtain an $O(1)$-approximation D^j to the best j-subspace using approximate volume sampling [22]. Then, $r = O(\varepsilon^{-2} \log 1/\delta)$ samples $s_1, \ldots, s_r$ are drawn with replacement from C, where $\Pr[C_i] = \frac{\text{dist}(C_i, D^j)}{\text{cost}(C, D^j)}$. Point s_i is assigned weight $\frac{1}{\Pr[s_i]}$. For each s_i, let $s_i' = \text{proj}(s_i, D^j)$, the projection of s_i onto D^j, which is assigned a weight of $-\frac{1}{\Pr[s_i]}$. Finally, all points are projected onto D^j. In recursive steps, an $O(1)$-approximation D^{j-1} to the best $j-1$-subspace of $\text{proj}(C, D^j)$ is found, and the above sampling procedure is repeated. The recursion stops when all points are projected to the origin. The weighted coreset is the union of the s_i and s_i' over the $j + 1$ stages. In [29], it is shown that for any fixed subspace F, the sum of (weighted) distances of coreset points to F is an unbiased estimator of $\text{cost}(C, F)$ and is an (ε, δ)-approximation.

While we have an NBE, and in this case making $\delta = \exp(-k)$, a sharply concentrated one, this construction is

not communication-efficient. We now describe our PAP for this problem assuming additively homomorphic encryption, which achieves $O^*(d^2)$ communication, $O^*(nd)$ work, and $O^*(1)$ rounds. We then show how to reduce the communication to near-optimal $O^*(d)$ assuming FHE.

We first show how to obtain shares of an $O(1)$-approximation D^j to the best j-subspace with probability $1-\exp(-k)$ using our sampling procedure to privately implement approximate volume sampling. Importantly, the actual D^j we obtain won't matter, as the estimator of [29] is an NBE and is sharply concentrated provided D^j is any $O(1)$-approximation. Hence, we will be able to apply the result of [47].

Consider the quantity $F_1(\ell_2(C)) = \sum_{i=1}^n \|C_i\|_2$. We first sample a row C_i with probability $\frac{\|C_i\|_2}{F_1(\ell_2(C))}$, using that an $O^*(1)$-communication and $O^*(nd)$-computation protocol for (ε,δ)-approximation to $F_1(\ell_2)$ exists [2]. We use SPIR to retrieve A_i, B_i, then compute $\|C_i\|_2$ exactly with $O^*(d)$ communication, which allows us to do rejection sampling to output a coin with bias $\frac{F_1(\ell_2(C))}{B}$ for an upper bound B. We repeatedly halve B until we obtain a sample C_{i_1}, i.e., until we do not reject. Then C_{i_1} is sampled with probability $\frac{\|C_i\|_2}{\sum_{i=1}^n \|C_i\|_2}$, and is additively shared. This is the same as our g-SAMPLER protocol, except it is applied to vectors.

An SFE then computes the $d \times d$ projection matrix P_1 corresponding to C_{i_1}, and sends the parties an additively homomorphic encryption $E(I-P_1)$, where I is the $d \times d$ identity matrix. The parties compute $E(A\cdot(I-P_1))$ and $E(B\cdot(I-P_1))$ using the homomorphism. The second crucial observation is that the sketch of [2] is a linear map, so it can be applied to the encryptions of the new points. We repeat this process, the SFE obtains C_{i_1}, C_{i_2}, and computes a homomorphic encryption of $I-P_2$, where P_2 is the projection onto $\text{span}\{C_{i_1}, C_{i_2}\}$, and the parties compute $E(A\cdot(I-P_2))$ and $E(B\cdot(I-P_2))$. The process repeats until the points are homomorphically encrypted on the orthogonal complement of $D^j = \text{span}\{C_{i_1}, C_{i_2}, \ldots, C_{i_j}\}$. The parties also compute homomorphic encryptions of their points projected onto D^j.

Given our approximate volume sampling, implementing [29] can again be done by sampling a homomorphically encrypted row according to its ℓ_2 norm using [2] (these rows are now the normal vectors to D^j). Inductively, the entire procedure of [29] can be implemented this way. Setting $\delta = \exp(-k)$, we get a sharply concentrated NBE and can appeal to [47]. The critical use of our transformation was to privately obtain a sample according to its ℓ_2-norm in an unbiased way. Our PAP generalizes to sampling rows (blocks) according to any norm (not just ℓ_2), using [2].

To achieve communication $O^*(d)$, note that the projection matrices P_i have rank at most $j = O(1)$, so can instead be communicated using FHE with $O^*(d)$ bits. There is an $\Omega(d)$ lower bound, which follows even to store a single point.

ℓ_p-sampling and Cascaded Moments. Sampling according to the distribution π is useful in its own right, for cascaded moments [3, 41, 48], machine learning problems (here $g(z) = z^2$) [17], and forward sampling [21, 48]. There are no known NBEs for these problems. Our importance sampling procedure solves this sampling primitive privately.

As an example application, the authors of [41] study estimating the cascaded moment $F_q(F_p(A))$ of an $n \times d$ matrix A, defined as $\sum_{i=1}^n (\sum_{j=1}^d |A_{i,j}|^p)^q$, for integer constants q, p and give a near-optimal $O^*(n^{1-2/(qp)}d^{1-2/p})$ space algorithm for integers $q \geq p \geq 2$. In [3], the authors obtained optimal space, up to $O^*(1)$ factors, for every q and p. To obtain a PAP, we first use our importance sampling procedure with a binary search on rows and black box use of the non-private algorithm of [3] to sample a row A_i with probability $r_i = C \cdot \frac{F_q(F_p(A_i))}{B}$, for a sufficiently large constant C (that depends on q and p) and an upper bound B on $F_q(F_p(A))$ (we can achieve any constant C with minor modifications to our g-SAMPLER protocol). The next observation is that $F_q(F_p(A_i))$ is a low-degree polynomial, namely, it equals $\prod_{j_1,\ldots,j_q} |A_{i,j_1} \cdots A_{j_q}|^p$. We use our importance sampling procedure with a non-private F_p-estimation algorithm to obtain samples $A_{i,j_1}, \ldots, A_{i,j_q}$ each with an approximation to their probability, denoted $\beta_1, \ldots, \beta_q$. Then $r_i \cdot \beta_1 \cdots \beta_q$ is our probability of sampling the monomial $A_{i,j_1} \ldots A_{i,j_q}$, which we can enforce is a constant-factor over-estimate to $\frac{|A_{i,j_1}\cdots A_{j_q}|^p}{F_q(F_p(A_i))}$. We can then compute $|A_{i,j_1} \cdots A_{j_q}|^p$ exactly, then reject the sampled monomial with the appropriate probability, so that, summing over all monomials, the probability we do not reject whatever sampled monomial we obtain equals $\frac{F_q(F_p(A_i))}{B}$. The protocol proceeds as in MAIN, by halving B, etc.

ℓ_2-Distance to Independence of Datasets. In [37], Indyk and McGregor study the streaming version of the problem: Alice has $(i,j,a_{i,j}) \in [n]^2 \times \{0,1,\ldots,M\}$, and Bob has $(i,j,b_{i,j}) \in [n]^2 \times \{0,1,\ldots,M\}$. Define the joint probabilities $p_{i,j} = \frac{a_{i,j}+b_{i,j}}{\sum_{i',j'} a_{i',j'}+b_{i',j'}}$, and marginals $q_i = \frac{\sum_{j'} a_{i,j'}+b_{i,j'}}{\sum_{i',j'} a_{i',j'}+b_{i',j'}}$ and $r_j = \frac{\sum_{i'} a_{i',j}+b_{i',j}}{\sum_{i',j'} a_{i',j'}+b_{i',j'}}$. They obtain an (ε,δ)-approximation for $h(a,b) = \sum_{i,j}(p_{i,j} - q_i r_j)^2$ in $O^*(1)$ space in $O^*(n^2)$ time. Their algorithm chooses independent 4-wise independent vectors $u, v \in \{-1,+1\}^n$, maintains $s = \sum_{i,j} u_i v_j (a_{i,j}+b_{i,j})$, $t_1 = \sum_i u_i \sum_j (a_{i,j}+b_{i,j})$, $t_2 = \sum_j v_j \sum_i (a_{i,j}+b_{i,j})$, and $L = \sum_{i',j'} a_{i',j'} + b_{i',j'}$, and computes $(\frac{s}{L} - \frac{t_1 t_2}{L^2})^2$. It averages out $O(\varepsilon^{-2})$ independent copies, and takes the median of $O(\log 1/\delta)$ independent averages. Their algorithm is not an NBE due to the median.

To obtain a PAP, we treat q, r, and L as fixed, coming from the outer level of recursion. Define $h(a,b,q,r,L) = \sum_{i,j}\left(\frac{a_{i,j}+b_{i,j}}{L} - q_i r_j\right)^2$. The key point is that the sketch of [37] provides an (ε,δ)-approximation even if p, q, and r are arbitrary vectors (of dimension n^2, n, and n, respectively). We sample an $i^* \in [n]$, expressing $h(a,b,q,r,L)$ as the quantity $\sum_i (\sum_j (\frac{a_{i,j}+b_{i,j}}{L} - q_i r_j)^2)$, and use binary search together with the sketch of [37] to get an $i^* \in [n]$ with probability $\frac{C}{B}\sum_j (\frac{a_{i^*,j}+b_{i^*,j}}{L} - q_{i^*} r_j)^2$ for an upper bound B on $h(a,b,q,r,L)$ and a $C > 1$ that can be computed. In our applications of the sketch of [37], we sum over all i,j in sketches t_1, t_2, and L above, but for s we only sum over the set of i corresponding to the current internal node of the binary tree (though we sum over all $j \in [n]$). Omitting details, we sample a coordinate j^* for this i^*, resulting in a pair (i^*, j^*) with probability $\frac{C'}{B}(\frac{a_{i^*,j^*}+b_{i^*,j^*}}{L} - q_{i^*} r_{j^*})^2$, for a value $C' > 1$ that can be computed, and an upper bound B on $h(a,b,q,r,L) = h(a,b)$. Via rejection sampling, we can flip a coin with bias $\frac{h(a,b,q,r,L)}{B}$, and proceed as usual.

Acknowledgments. I thank Alexandre Evfimievski, Ronald Fagin, Vitaly Feldman, Yuval Ishai, Ilya Mironov, and the anonymous referees for helpful comments.

5. REFERENCES

[1] N. Alon, Y. Matias, and M. Szegedy. The Space Complexity of Approximating the Frequency Moments. *JCSS*, 58(1):137–147, 1999.

[2] A. Andoni, K. D. Ba, P. Indyk, and D. P. Woodruff. Efficient sketches for earth-mover distance, with applications. In *FOCS*, pages 324–330, 2009.

[3] A. Andoni, R. Krauthgamer, and K. Onak. Streaming Algorithms via Precision Sampling. *CoRR*, abs/1011.1263, 2010.

[4] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *JCSS*, 68(4):702–732, 2004.

[5] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *SODA*, pages 623–632, 2002.

[6] A. Beimel, P. Carmi, K. Nissim, and E. Weinreb. Private approximation of search problems. In *STOC*, pages 119–128, 2006.

[7] A. Beimel, R. Hallak, and K. Nissim. Private approximation of clustering and vertex cover. *Computational Complexity*, 18(3):435–494, 2009.

[8] L. Bhuvanagiri and S. Ganguly. Estimating entropy over data streams. In *ESA*, pages 148–159, 2006.

[9] L. Bhuvanagiri, S. Ganguly, D. Kesh, and C. Saha. Simpler algorithm for estimating frequency moments of data streams. In *SODA*, pages 708–713, 2006.

[10] V. Braverman and R. Ostrovsky. Zero-one frequency laws. In *STOC*, 2010.

[11] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *EUROCRYPT*, pages 402–414, 1999.

[12] A. R. Calderbank, A. C. Gilbert, K. Levchenko, S. Muthukrishnan, and M. Strauss. Improved range-summable random variable construction algorithms. In *SODA*, pages 840–849, 2005.

[13] R. Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 13(1):143–202, 2000.

[14] A. Chakrabarti, G. Cormode, and A. McGregor. A near-optimal algorithm for computing the entropy of a stream. In *SODA*, pages 328–335, 2007.

[15] A. Chakrabarti, K. Do Ba, and S. Muthukrishnan. Estimating Entropy and Entropy Norm on Data Streams. In *STACS*, pages 196–205, 2006.

[16] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *ICALP*, pages 693–703, 2002.

[17] K. L. Clarkson, E. Hazan, and D. Woodruff. Sublinear optimization for machine learning. In *FOCS*, 2010.

[18] G. Cormode, M. Datar, P. Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Trans. Knowl. Data Eng.*, 15(3):529–540, 2003.

[19] G. Cormode and S. Muthukrishnan. Estimating dominance norms of multiple data streams. In *ESA*, pages 148–160, 2003.

[20] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.

[21] G. Cormode, S. Muthukrishnan, and I. Rozenbaum. Summarizing and mining inverse distributions on data streams via dynamic inverse sampling. In *VLDB*, pages 25–36, 2005.

[22] A. Deshpande and K. R. Varadarajan. Sampling-based dimension reduction for subspace approximation. In *STOC*, pages 641–650, 2007.

[23] A. Deshpande, K. R. Varadarajan, M. Tulsiani, and N. K. Vishnoi. Algorithms and hardness for subspace approximation. *CoRR*, abs/0912.1403, 2009.

[24] C. Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[25] C. Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1–19, 2008.

[26] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. J. Strauss, and R. N. Wright. Secure multiparty computation of approximations. *ACM Transactions on Algorithms*, 2(3):435–472, 2006.

[27] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate L1-difference algorithm for massive data streams. *SIAM J. Comput.*, 32(1):131–151, 2002.

[28] D. Feldman, A. Fiat, and M. Sharir. Coresets forweighted facilities and their applications. In *FOCS*, pages 315–324, 2006.

[29] D. Feldman, M. Monemizadeh, C. Sohler, and D. P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *SODA*, pages 630–649, 2010.

[30] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[31] C. Gentry and Z. Ramzan. Single-database private information retrieval with constant communication rate. In *ICALP*, pages 803–815, 2005.

[32] O. Goldreich. Secure multi-party computation. 2000.

[33] S. Guha, A. McGregor, and S. Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *SODA*, pages 733–742, 2006.

[34] S. Halevi, R. Krauthgamer, E. Kushilevitz, and K. Nissim. Private approximation of np-hard functions. In *STOC*, pages 550–559, 2001.

[35] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *STOC*, pages 291–300, 2004.

[36] N. J. A. Harvey, J. Nelson, and K. Onak. Sketching and streaming entropy via approximation theory. In *FOCS*, pages 489–498, 2008.

[37] P. Indyk and A. McGregor. Declaring independence via the sketching of sketches. In *SODA*, 2008.

[38] P. Indyk and D. P. Woodruff. Optimal approximations of the frequency moments of data streams. In *STOC*, 2005.

[39] P. Indyk and D. P. Woodruff. Polylogarithmic private approximations and efficient matching. In *TCC*, pages 245–264, 2006.

[40] Y. Ishai, T. Malkin, M. J. Strauss, and R. N. Wright. Private multiparty sampling and approximation of vector combinations. *Theor. Comput. Sci.*, 410(18):1730–1745, 2009.

[41] T. S. Jayram and D. P. Woodruff. The data stream space complexity of cascaded norms. In *FOCS*, 2009.

[42] D. M. Kane, J. Nelson, E. Porat, and D. P. Woodruff. Fast moment estimation in data streams in optimal space. In *STOC*, 2011, to appear.

[43] D. M. Kane, J. Nelson, and D. P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *SODA*, pages 1161–1178, 2010.

[44] J. Kilian, A. Madeira, M. J. Strauss, and X. Zheng. Fast private norm estimation and heavy hitters. In *TCC*, pages 176–193, 2008.

[45] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[46] P. Li. Estimators and tail bounds for dimension reduction in l_p $(0 < p \leq 2)$ using stable random projections. In *SODA*, pages 10–19, 2008.

[47] A. Madeira and S. Muthukrishnan. Functionally private approximations of negligibly-biased estimators. In *FSTTCS*, pages 323–334, 2009.

[48] M. Monemizadeh and D. P. Woodruff. 1-pass relative-error l_p-sampling with applications. In *SODA*, 2010.

[49] S. Muthukrishnan. Data Streams: Algorithms and Applications. *Foundations and Trends in Theoretical Computer Science*, 1(2):117–236, 2005.

[50] M. Naor and K. Nissim. Communication preserving protocols for secure function evaluation. In *STOC*, pages 590–599, 2001.

[51] M. Naor and B. Pinkas. Oblivious polynomial evaluation. *SIAM J. Comput.*, 35(5):1254–1281, 2006.

[52] J. Nelson and D. P. Woodruff. Fast manhattan sketches in data streams. In *PODS*, pages 99–110, 2010.

[53] A. Pavan and S. Tirthapura. Range-efficient counting of distinct elements in a massive data stream. *SIAM J. Comput.*, 37(2):359–379, 2007.

[54] M. E. Saks and X. Sun. Space lower bounds for distance approximation in the data stream model. In *STOC*, pages 360–369, 2002.

[55] N. D. Shyamalkumar and K. R. Varadarajan. Efficient subspace approximation algorithms. In *SODA*, 2007.

[56] S. Stoev, M. Hadjieleftheriou, G. Kollios, and M. S. Taqqu. Norm, point, and distance estimation over multiple signals using max-stable distributions. In *ICDE*, 2007.

[57] S. Stoev and M. S. Taqqu. Max-stable sketches: estimation of lp-norms, dominance norms and point queries for non-negative signals. *CoRR*, abs/1005.4344, 2010.

[58] H. Sun and C. K. Poon. Two improved range-efficient algorithms for f_0 estimation. *Theor. Comput. Sci.*, 410(11):1073–1080, 2009.

[59] M. Thorup and Y. Zhang. Tabulation based 4-universal hashing with applications to second moment estimation. In *SODA*, pages 615–624, 2004.