Distributed Sketching for Randomized Optimization: Exact Characterization, Concentration and Lower Bounds

Burak Bartan and Mert Pilanci, *Member, IEEE*

***Abstract*—We consider distributed optimization methods for problems where forming the Hessian is computationally challenging and communication is a significant bottleneck. We leverage randomized sketches for reducing the problem dimensions as well as preserving privacy and improving straggler resilience in asynchronous distributed systems. We derive novel approximation guarantees for classical sketching methods and establish tight concentration results that serve as both upper and lower bounds on the error. We then extend our analysis to the accuracy of parameter averaging for distributed sketches. Furthermore, we develop unbiased parameter averaging methods for randomized second order optimization in regularized problems that employ sketching of the Hessian. Existing works do not take the bias of the estimators into consideration, which limits their application to massively parallel computation. We provide closed-form formulas for regularization parameters and step sizes that provably minimize the bias for sketched Newton directions. Additionally, we demonstrate the implications of our theoretical findings via large scale experiments on a serverless cloud computing platform.**

***Index Terms*—sketching, distributed optimization, randomized algorithms, convex optimization, regularized least squares, second order optimization, large scale problems, differential privacy**

I. Introduction

We investigate distributed sketching methods for solving large scale optimization problems, including regularized linear regression as a special case. A standard model that we consider in this work is the least squares problem given by

$$x^* = \arg\min_x \|Ax - b\|_2^2 + \lambda \|x\|_2^2 \tag{1}$$

where $A \in \mathbb{R}^{n \times d}$ is the data matrix and $b \in \mathbb{R}^n$ is the target vector. Here, $\lambda \in \mathbb{R}$ is the coefficient for the squared ℓ_2 norm regularization on the parameters $x \in \mathbb{R}^d$. We consider the setting where the data matrix A is large and thus distributing the computation among multiple computing nodes is desirable. We consider both underdetermined and overdetermined linear regression problems in the regime where the data does not fit in main memory. Such linear regression problems and linear systems are commonly encountered in a multitude of problems ranging from statistics and machine learning to optimization.

B. Bartan is with the Department of Electrical Engineering, Stanford University, CA, 94305 USA (e-mail: bbartan@stanford.edu).

M. Pilanci is with the Department of Electrical Engineering, Stanford University, CA, 94305 USA (e-mail: pilanci@stanford.edu).

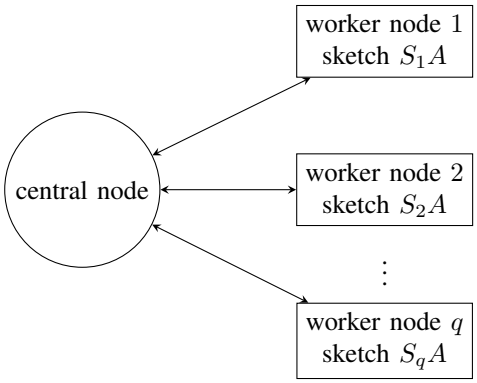


Fig. 1. Distributed computing model. There is a single central node and q worker nodes. The worker nodes only communicate with the central node.

Being able to solve large scale linear regression problems efficiently is crucial for many applications.

We focus on a centralized distributed computing model. Namely, we assume a single central node and multiple worker nodes that operate in parallel. This model is visualized in Figure 1. The worker nodes only communicate with the central node, i.e., no communication among worker nodes is assumed. The algorithms that we study in this work typically involve worker nodes computing an estimate solution based on a sketch or subsampled form of the data, which is then communicated to the central node. The central node averages the estimates from the worker nodes. We will establish theoretical results for the approximation error of the resulting estimate.

In the proposed scheme, the workers compute their local estimates by solving smaller subproblems. The way the subproblems are constructed is through random sketches or subsamples. Rather than considering the entire data A, b, worker nodes perform computations on sketched data SA, Sb where $S \in \mathbb{R}^{m \times n}$ is a random sketching matrix with $m \ll n$. Applications of randomized sketches and dimension reduction to linear regression and other optimization problems have been extensively studied in the recent literature including [1], [2], [3], [4], [5], [6]. In this work, we investigate averaging the solutions of sketched sub-problems. The setting for overdetermined problems was also studied in [2]. In addition, we consider underdetermined regression problems where the number of data samples is less than the dimension of the unknown parameter and investigate the properties of averaging for such problems.

We extend our results to problems beyond least squares regression through iterative randomized optimization methods including Newton Sketch [7], [3] and introduce their distributed variants. We focus on the communication-efficient setting where we avoid the communication of approximate Hessian matrices of size d^2 and communicate only the approximate solutions of size d, where d is the parameter dimension. Averaging sketched solutions was proposed in the literature in certain restrictive settings [8]. The presence of a regularization term requires a more detailed analysis, as we will show that naïve averaging leads to biased estimators of the solution (see Theorem 4.3). We note that this bias is with respect to the randomness of the sketching matrices; the input data are not assumed to be sampled from a probability distribution. Although the bias is often overlooked in the literature, we show that one can re-calibrate the regularization coefficient of the sketched problems to obtain unbiased estimators. We show that having unbiased estimators leads to better performance without imposing any additional computational cost. Furthermore, we show that our bias correction method empirically works for many other non-Gaussian sketches, including row sampling.

Another important advantage of the proposed scheme is in asynchronous distributed computing. Employing parameter averaging enables asynchronous updates, since a running average of available parameters can approximate the result without requiring all worker nodes to finish their tasks. In other words, an important advantage of randomized sketching in distributed computing is the independent and identically distributed nature of all of the computational tasks. Therefore, distributed sketching offers a resilient computing model where node failures and stragglers as well as additions of new nodes can be easily handled, e.g., via generating additional data sketches. An alternative to averaging that would offer similar benefits is the asynchronous stochastic gradient descent (SGD) algorithm [9], [10]. However, the convergence rates of asynchronous SGD methods necessarily depend on the properties of the input data such as its condition number [9], [10]. In contrast, as we show in this work, distributed sketching has stronger convergence guarantees that do not depend on the condition number of the data matrix. Moreover, sketching provably preserves the privacy of the data (see Section V), making it an attractive choice for massively parallel cloud computing.

Although the main focus of this paper is on distributed computing and parameter averaging, we provide improved theoretical results for classical, i.e., non-distributed, sketching. Novel solution approximation results including exact characterizations of the expected error and exponential concentration are derived for the single sketch estimator as well. Table I summarizes the classes of problems studied in this work. The last column of the table contains references to the theorems and lemmas for each result. We first focus on Gaussian sketches for the ease of exposition and leave extensions to other sketches (e.g., row sampling, leverage sampling and Hadamard) to the appendix.

A. Cloud Computing

The methods considered in this paper can be used various distributed computing environments, including both conventional server-based systems and serverless systems. Serverless computing is a relatively new approach to distributed systems that offers computing on the cloud without requiring any server management from end users [11], [12], [13]. Functions in serverless computing can be thought of worker nodes which have very limited resources and a lifetime, but also are very scalable. It is possible to launch thousands of serverless computing jobs within a few seconds. The methods that we introduce in this paper are particularly suitable for serverless computing platforms, since the algorithms do not require peer-to-peer communication among different worker nodes, and have low memory and compute requirements per node. In addition, we present novel exact expressions for the expected error of distributed averaging estimators. These are most useful when the number of nodes q in the distributed computing system is large, which is readily achieved by serverless computing. In Section VII, we provide a large scale evaluation of our methods on the serverless computing platform AWS Lambda.

Data privacy is an increasingly important issue in cloud computing, one that has been studied in many recent works including [14], [15], [16], [17]. A remarkable benefit of distributed sketching methods we consider is privacy preservation. To be more precise, let us consider a setting where the center node computes sketched data S_kA, S_kb, $k = 1, \ldots, q$ locally where $S_k \in \mathbb{R}^{m\times n}$ are the sketching matrices and $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$ are the data matrix and the output vector, respectively, for the regression problem $\min_x \|Ax-b\|_2^2$. In the distributed sketching setting, the central node sends only the sketched data to worker nodes for computational efficiency, as well as data privacy preservation. In particular, the mutual information and differential privacy can be controlled when we reveal S_kA and keep A hidden. Furthermore, one can trade privacy for accuracy by choosing a suitable sketch dimension m (see Theorem 5.2).

B. Notation

We use hats $\hat{x}$ to denote the estimator for a single sketch and bars $\bar{x}$ to denote the averaged estimator $\bar{x} := \frac{1}{q}\sum_{k=1}^q \hat{x}_k$, where $\hat{x}_k$ is the estimator for the k'th worker node. Stars are used to denote the optimal solution x^*. The data matrix and target vector are denoted by $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, respectively. We use $f(\cdot)$ to denote the objective of the optimization problem being considered at that point in the text. The letter ϵ is used for error while ε is used as the differential privacy parameter. For instance, to analyze the concentration of the approximation error, we typically show high probability bounds for the event $\left|\frac{f(\bar{x})}{f(x^*)} - \mathbb{E}\,\frac{f(\bar{x})}{f(x^*)}\right| < \epsilon$. The notation $\sigma_{\min}(\cdot)$ denotes the smallest nonzero singular value of its argument. $O(\cdot)$ is used for the big-O notation.

All the expectations in the paper are with respect to the randomness over the sketching matrices, and no randomness assumptions are made for the data. We use $S \in \mathbb{R}^{m\times n}$ to denote random sketching matrices and we often refer to m

TABLE I
SUMMARY OF THEORETICAL RESULTS. THE DATA MATRIX IS $A \in \mathbb{R}^{n\times d}$ AND THE OUTPUT VECTOR IS $b \in \mathbb{R}^n$.

Problem	Sketch Type	Method and Theorem
$\min_x \|Ax-b\|_2^2$	Gaussian	Distributed randomized regression: Theorem 2.2, 2.8, 2.11
	Gaussian	Distributed Iterative Hessian sketch: Theorem 3.2
	Other	Distributed randomized regression: Lemma A.3, A.4, A.5, Theorem A.6
$\min_x \|x\|_2^2$ s.t. $Ax=b$	Gaussian	Right sketch: Theorem 2.13
$\min_x \|Ax-b\|_2^2+\lambda_1\|x\|_2^2$	Gaussian	Distributed randomized ridge regression: Theorem 4.3
Convex problems with Hessian $(H_t^{1/2})^T H_t^{1/2}$	Gaussian	Distributed Newton sketch: Theorem 3.5 (step size for unbiasedness)
Convex problems with Hessian $(H_t^{1/2})^T H_t^{1/2}+\lambda_1 I_d$	Gaussian	Distributed Newton sketch: Theorem 4.5 (regularization coefficient for unbiasedness)

as the sketch size. For non-iterative distributed algorithms, we use $S_k \in \mathbb{R}^{m\times n}$ for the sketching matrix used by worker node k. For iterative algorithms, $S_{t,k} \in \mathbb{R}^{m\times n}$ is used to denote the sketching matrix used by worker k in iteration t. We assume the sketching matrices are scaled such that $\mathbb{E}[S^TS]=I_n$. We omit the subscripts in S_k and $S_{t,k}$ for simplicity whenever it does not cause confusion. The notation $m \gtrsim d$ is used to denote that there exists a positive finite constant c such that $m \geq cd$. We denote the thin SVD decomposition of the data matrix as $A=U\Sigma V^T$.

For regularized problems, we use λ_1 for the regularization coefficient of the original problem, and λ_2 for the regularization coefficient of the sketched problem. For instance, in the case of the regularized least squares problem, we have

$$x^* = \arg\min_x \|Ax-b\|_2^2+\lambda_1\|x\|_2^2\,,$$
$$\hat{x} = \arg\min_x \|SAx-Sb\|_2^2+\lambda_2\|x\|_2^2\,. \tag{2}$$

We will derive expressions in the sequel for the optimal selection of the coefficient λ_2 which we denote as λ_2^*.

C. Related Work

Random projections are a popular way of performing randomized dimensionality reduction, which are widely used in many computational and learning problems [18], [5], [19], [20]. Many works have studied randomized sketching methods for least squares and other optimization problems [21], [22], [1], [6], [7], [2], [23].

Sarlós [24] showed that the relative error of the single Gaussian sketch estimator in (2) in the unregularized case $\lambda_1 = \lambda_2 = 0$ is bounded as $f(\hat{x})/f(x^*) \leq (1+Cd\log(d)/m)^2$ with probability at least $1/3$, where C is a constant. The relative error was improved to $f(\hat{x})/f(x^*) \leq (1+Cd/m)$ with exponentially small failure probability in subsequent work [6]. In contrast, we derive the expectation of the error exactly and show exponential concentration around this expected value in this paper. As a result, we obtain a tight upper and lower bound for the relative error that holds with exponentially high probability (see Theorem 2.8).

The work [2] investigates distributed sketching and averaging for regression from optimization and statistical perspectives. The most relevant result in [2], using our notation, can be stated as follows. Setting the sketch dimension $m = O(\mu d(\log d)/\epsilon)$ for uniform sampling, where μ is row coherence $\mu := n/\operatorname{rank}(A)\max_i \ell_i$ and ℓ_i is the i'th row leverage score, and $m=\tilde{O}(d/\epsilon)$ (with $\tilde{O}$ hiding logarithmic factors) for other sketches, the inequality $f(\bar{x})-f(x^*) \leq (\epsilon/q+\epsilon^2)f(x^*)$ holds with high probability. According to this result, for large q, the cost at the averaged solution $f(\bar{x})$ will be upper bounded by $\epsilon^2 f(x^*)$. In this work we prove that for Gaussian sketch, $f(\bar{x})$ will converge to the optimal cost $f(x^*)$ as q increases. We also identify the exact expected error for a given number of workers q. In addition, our results on regularized least squares regression improve on the results in [2] for averaging multiple sketched solutions. In particular, in [2], the sketched sub-problems use the same regularization parameter as the original problem, which leads to biased solutions. We analyze the bias of the averaged solution, and provide explicit formulas for selecting the regularization parameter of the sketched sub-problems to achieve unbiasedness.

We show that the expected difference between the costs of the averaged solution and the optimal solution has two components, namely variance and squared bias (see Lemma A.1). This result implies that for the Gaussian sketch, which we prove to be unbiased (see Lemma 2.1), the number of workers required for a target error ϵ scales as $1/\epsilon$. Remarkably, this result does not involve condition numbers. In contrast, for the asynchronous Hogwild algorithm [10], the number of iterations required for error ϵ scales with $\log(1/\epsilon)/\epsilon$ and also depends on the condition number of the input data, although it addresses a more general class of problems.

One of the crucial results of our work is developing bias correction for averaging sketched solutions. Namely, the setting considered in [23] is based on a distributed second order optimization method that involves averaging approximate update directions. However, in that work, the bias was not taken into account which degrades accuracy and limits applicability to massively parallel computation. We additionally provide results for the unregularized case, which corresponds to the distributed version of the Newton sketch algorithm introduced in [7].

D. Overview of Our Contributions

- We characterize the exact expected error of the averaged estimator for Gaussian sketch in closed form as

$$\frac{\mathbb{E}[f(\bar{x})]-f(x^*)}{f(x^*)} = \frac{1}{q}\frac{d}{m-d-1} \tag{3}$$

where $\bar{x}=\frac{1}{q}\sum_{k=1}^q \hat{x}_k$ is the averaged solution and $\hat{x}_k = \arg\min_x \|S_kAx-S_kb\|_2^2$ is the output of the k'th worker

node. In addition, we obtain a similar result for the error of the averaged estimator for the least-norm solution in the underdetermined case $n < d$.

- For both the single sketch and averaged estimator, we show that the relative error $\frac{f(\bar{x})-f(x^*)}{f(x^*)}$ is concentrated around its expectation given in (3) with exponentially high probability. This provides both an upper and lower bound on the performance of sketching, unlike previous results in the literature. Moreover, it offers a guaranteed recipe to set the sketch size m and number of workers q.
- We show that for Gaussian distributed sketch, the expected error of the averaged estimator $\mathbb{E}[f(\bar{x})] - f(x^*)$ matches the error lower bound for any unbiased estimator obtained via Fisher information. In addition, we provide a lower bound for general, possibly biased sketching based estimators.
- We consider the privacy preserving properties of distributed sketching methods, in which only random projections of the data matrix A need to be shared with worker nodes. We derive conditions on the (ε, δ)-differential privacy when S is the i.i.d. Gaussian sketch matrix. Combined with our results, we show that the approximation error of this distributed privacy preserving regression algorithm scales as $O(1/\varepsilon^2)$.
- We analyze the convergence rate of a distributed version of the iterative Hessian sketch algorithm which was introduced in [3] and show that the number of iterations required to reach error ϵ with q workers scales with $\log(1/\epsilon)/\log(q)$.
- We show that $\hat{x} = \arg\min_x \|SAx - Sb\|_2^2 + \lambda_2\|x\|_2^2$ is not an unbiased estimator of the optimal solution, i.e. $\mathbb{E}[A(\hat{x} - x^*)] \neq 0$ when $\lambda_2 = \lambda_1$ and provide a closed form expression for the regularization coefficient λ_2^* to make the estimator unbiased under certain assumptions.
- In addition to Gaussian sketch, we derive bias bounds for uniform sampling, randomized Hadamard sketch and leverage score sampling. Analysis of the bias term is critical in understanding how close to the optimal solution we can hope to get and establishing the dependence on the sketch dimension m. Moreover, we utilize the derived bias bounds and find an upper bound on the error of the averaged estimator for these other sketching methods.
- We discuss averaging for the distributed version of Newton Sketch [7] and show that using the same regularization coefficient as the original problem, i.e., $\lambda_2 = \lambda_1$, as most works in the literature consider, is not optimal. Furthermore, we derive an expression for choosing the regularization coefficient λ_2^* for unbiasedness.
- We provide numerical simulations that illustrate the practicality and scalability of distributed sketching methods on the serverless cloud computing platform AWS Lambda.

E. Preliminaries on Sketching Matrices

We consider various sketching matrices in this work including Gaussian sketch, uniform sampling, randomized Hadamard sketch, Sparse Johnson-Lindenstrauss Transform (SJLT), and *hybrid* sketch. We now briefly describe each of these sketching methods:

1) *Gaussian sketch [24]:* Entries of $S \in \mathbb{R}^{m\times n}$ are i.i.d. and sampled from the Gaussian distribution. Sketching a matrix $A \in \mathbb{R}^{n\times d}$ using Gaussian sketch requires computing the matrix product SA which has computational complexity equal to $O(mnd)$.
2) *Randomized Hadamard sketch [25]:* The sketch matrix in this case can be represented as $S = PHD$ where $P \in \mathbb{R}^{m\times n}$ is for uniform sampling of m rows out of n rows, $H \in \mathbb{R}^{n\times n}$ is the Hadamard matrix, and $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix with diagonal entries sampled randomly from the Rademacher distribution. Multiplication by D to obtain DA requires $O(nd)$ scalar multiplications. Hadamard transform can be implemented as a fast transform with complexity $O(n\log(n))$ per column, and a total complexity of $O(nd\log(n))$ to sketch all d columns of DA.
3) *Uniform sampling [26]:* Uniform sampling randomly selects m rows out of the n rows of A where the probability of any row being selected is the same.
4) *Leverage score sampling:* Row leverage scores of a matrix A are given by $\ell_i = \|\tilde{u}_i\|_2^2$ for $i = 1, \ldots, n$ where $\tilde{u}_i \in \mathbb{R}^d$ denotes the i'th row of U. The matrix U is the matrix whose columns are the left singular vectors of A, i.e., $A = U\Sigma V^T$. There is only one nonzero element in every row $s_i \in \mathbb{R}^n$ of the sketching matrix S and the probability that the j'th entry of s_i is nonzero is proportional to the leverage score ℓ_j. More precisely, the rows $s_1, \ldots, s_m$ are sampled i.i.d. such that $\mathbb{P}[s_i = e_k/\sqrt{mp_k}] = p_k$, $\forall i$, $\forall k$ where $p_k = \frac{\ell_k}{\sum_{j=1}^n \ell_j}$. Note that we have $\mathbb{E}[S^TS] = \mathbb{E}[\sum_{i=1}^m s_i s_i^T] = m\sum_{k=1}^n p_k e_k e_k^T/(mp_k) = I_n$. Naïve algorithm for computing leverage scores runs in $O(nd^2)$ time while the approximation algorithm in [27] runs in $O(nd\log(n))$ time.
5) *Sparse Johnson-Lindenstrauss Transform (SJLT) [28]:* The sketching matrix for SJLT is a sparse matrix where each column has exactly s nonzero entries and the columns are independently distributed. The nonzero entries are sampled from the Rademacher distribution. It takes $O(snd/m)$ addition operations to sketch a data matrix using SJLT.
6) *Hybrid sketch:* The method that we refer to as hybrid sketch is a sequential application of two different sketching methods. In particular, it might be computationally feasible for worker nodes to sample as much data as possible, say m' rows, and then reduce the dimension of the available data to the final sketch dimension m using another sketch with better error properties than uniform sampling such as Gaussian sketch or SJLT. For instance, a hybrid sketch of uniform sampling followed by Gaussian sketch would have computational complexity $O(mm'd)$.

F. Paper Organization

Section II deals with the application of distributed sketching to quadratic problems for Gaussian sketch. The algorithms in Section II are non-iterative. In Section III we consider distributed sketching algorithms for iterative algorithms and show how these ideas are applied to second order optimization for unconstrained convex problems. In Section IV, we present our results on bias correction for randomized sketching in regularized problems. Section V deals with the privacy preserving property of distributed sketching. Section VI provides an overview of applications and example problems where distributed sketching methods could be applied. Section VII presents numerical results and Section VIII concludes the main part of the paper. Section A of the appendix gives theoretical results for randomized Hadamard sketch, uniform sampling, and leverage score sampling. We give the proofs for the majority of the lemmas and theorems in the appendix along with additional numerical results.

II. Distributed Sketching for Quadratic Optimization problems

In this section, we focus on the regularized Least Squares optimization problem

$$x^* = \arg\min_x \|Ax - b\|_2^2 + \lambda_1 \|x\|_2^2 \,. \tag{4}$$

We study various distributed algorithms for solving problems of this form based on model averaging. Some of these algorithms are tailored for the unregularized case $\lambda_1 = 0$.

A. Closed Form Expressions for the Expected Error of Gaussian Sketch

We begin with a non-iterative model averaging based algorithm, which we refer to as distributed randomized regression. This algorithm is for the unregularized case $\lambda_1 = 0$. Each of the worker nodes computes an approximate solution $\hat{x}_k$ and these are averaged at the master node to compute the final solution $\bar{x}$. This method is outlined in Algorithm 1. The theoretical analysis in this section assumes that the sketch matrix is Gaussian, which is generalized to other sketching matrices in Section A. Although computing the Gaussian sketch is not as efficient as other fast sketches such as the randomized Hadamard sketch, it has several significant advantages over other sketches: (1) exact relative error expressions can be derived as we show in this section, (2) the solution is unbiased, (3) a differential privacy bound can be provided as we show in Section V, and (4) computation of the sketch can be trivially parallelized.

We first obtain a characterization of the expected error for the single sketch estimator in Lemma 2.1.

Lemma 2.1: For the Gaussian sketch with $m > d+1$, the estimator $\hat{x}$ satisfies

$$\mathbb{E}[\|A(\hat{x} - x^*)\|_2^2] = \mathbb{E}[f(\hat{x})] - f(x^*) = f(x^*)\frac{d}{m-d-1}\,, \tag{5}$$

where $f(x) := \|Ax - b\|_2^2$ and $x^* = \arg\min_x f(x)$.

Algorithm 1: Distributed Randomized Regression

Input: Data matrix $A \in \mathbb{R}^{n\times d}$, target vector $b \in \mathbb{R}^n$.
for worker $k = 1, \ldots, q$ in parallel **do**
  Obtain the sketched data and sketched output: $S_k A$ and $S_k b$.
  Solve $\hat{x}_k = \arg\min_x \|S_k A x - S_k b\|_2^2$ and send $\hat{x}_k$ to the master node.
end for
Master node: return $\bar{x} = \frac{1}{q}\sum_{k=1}^q \hat{x}_k$.
(*Master node: return* $\bar{x} = \frac{1}{|\mathcal{A}|}\sum_{k\in\mathcal{A}} \hat{x}_k$, *optional)*

Proof of Lemma 2.1: Suppose that the matrix A is full column rank. Then, for $m \geq d$, the matrix $A^T S^T S A$ follows a Wishart distribution, and is invertible with probability one. Conditioned on the invertibility of $A^T S^T S A$, we have

$$\begin{aligned}\hat{x} &= (A^T S^T S A)^{-1} A^T S^T S b \\ &= (A^T S^T S A)^{-1} A^T S^T S (A x^* + b^\perp) \\ &= x^* + (A^T S^T S A)^{-1} A^T S^T S b^\perp \,,\end{aligned}$$

where we have defined $b^\perp := b - A x^*$. Note that SA and $Sb^\perp$ are independent since they are Gaussian and uncorrelated as a result of the normal equations $A^T b^\perp = 0$. Conditioned on the realization of the matrix SA and the event $A^T S^T S A \succ 0$, a simple covariance calculation shows that

$$\hat{x} \sim \mathcal{N}\big(x^*, \frac{1}{m} f(x^*)(A^T S^T S A)^{-1}\big)\,. \tag{6}$$

Multiplying with the data matrix A on the left yields the distribution of the prediction error, conditioned on SA, as

$$A(\hat{x} - x^*) \sim \mathcal{N}\big(0, \frac{1}{m} f(x^*) A (A^T S^T S A)^{-1} A^T\big)\,. \tag{7}$$

Then we can compute the conditional expectation of the squared norm of the error

$$\mathbb{E}[\|A(\hat{x} - x^*)\|_2^2 \;\big|SA] = \frac{f(x^*)}{m}\,\mathbb{E}[\mathrm{tr}(A(A^T S^T S A)^{-1} A^T)]\,.$$

Next we recall that the expected inverse of the Wishart matrix $A^T S^T S A$ satisfies (see, e.g., [29])

$$\mathbb{E}[(A^T S^T S A)^{-1}] = (A^T A)^{-1}\frac{m}{m-d-1}\,.$$

Plugging in the previous result and using the tower property of expectations and then noting that $\mathrm{tr}(A(A^T A)^{-1} A^T) = d$ give us the claimed result. ∎

To the best of our knowledge, this result of exact error characterization is novel in the theory of sketching. Similar tools to those that we used in this proof were used in [30], however, the exact context in which they are used and the end result are different from this work. The expected error of the single sketch estimator appears in [31] as well. Furthermore, existing results (see e.g. [5], [2], [19]) characterize a high probability upper bound on the error, whereas the above is a sharp and exact formula for the expected squared norm of the error. Theorem 2.2 builds on Lemma 2.1 to characterize the expected error for the averaged solution $\bar{x}$.

Theorem 2.2 (Expected error of the averaging estimator): Let S_k, $k = 1, \dots, q$ be Gaussian sketching matrices, then Algorithm 1 runs in time $O(md^2)$, and the error of the averaged solution $\bar{x}$ satisfies

$$\frac{\mathbb{E}[f(\bar{x})] - f(x^*)}{f(x^*)} = \frac{1}{q}\frac{d}{m-d-1}\,. \tag{8}$$

Consequently, Markov's inequality implies that $\frac{f(\bar{x})-f(x^*)}{f(x^*)} \leq \epsilon$ holds with probability at least $\left(1 - \frac{1}{q\epsilon}\frac{d}{m-d-1}\right)$ for any $\epsilon > 0$.

Theorem 2.2 illustrates that the expected error scales as $1/q$, and converges to zero as $q \to \infty$ as long as $m \geq d+2$. This is due to the unbiasedness of the Gaussian sketch. Other sketching methods such as uniform sampling or randomized Hadamard sketch do not have this property, as we further investigate in Section A.

Remark 2.3: In Algorithm 1, the worker nodes are tasked with obtaining the sketched data S_kA and the sketched output S_kb. We identify two options for this step:

- Option 1: The master node computes the sketched data S_kA, S_kb, $k = 1, \dots, q$ and transmits to worker nodes. This option preserves data privacy, which we discuss in Section V.
- Option 2: The worker nodes have access to the data A, b and compute the sketched data S_kA and S_kb. This option does not preserve privacy, however, parallelizes the computation of the sketch via the workers.

It provides insight to compare the result in Theorem 2.2 against the baseline where the master node computes a sketch of size mq. Note that the error performance of the single big sketch would in fact outperform the distributed sketch case. The error of the single sketch is $\frac{d}{mq-d-1}$ while the error of the distributed averaging case is $\frac{1}{q}\frac{d}{m-d-1}$. Since $\frac{d}{mq-d-1} \leq \frac{d}{mq-dq-q}$, the error of the single big sketch case is smaller. It is important to point out that when $m \gg d$, the errors are asymptotically the same, i.e., $O(\frac{d}{mq})$. In addition, there are many advantages for the distributed averaging case over the single sketch case including parallelization of the computations of sketches and sub-problems, lower cost subproblems per worker node, and hence faster run time. In the single sketch case, solving the corresponding linear system is of complexity $O(mqd^2)$, while it is $O(md^2)$ per node in the distributed sketch case.

We note that Algorithm 1 is robust against straggling nodes and failures since the final output does not rely on any specific set of node outputs. This is reflected in the last line of Algorithm 1 where we average only the available set of node outputs $\mathcal{A}$. The observation here is that rather than waiting for the output of all q nodes, we can simply collect the available set of outputs and average them. The algorithms given in the remainder of the paper satisfy this property as well since they are all based on averaging i.i.d. node estimates.

B. Exponential Concentration of the Gaussian Sketch Estimator

In this subsection, we show high probability concentration bounds for the error of the Gaussian sketch. Importantly, our result provides both an upper and lower bounds on the relative error with exponentially small failure probability. We begin with the single sketch estimator $\hat{x}$ and extend the results to the averaged estimator $\bar{x} = \frac{1}{q}\sum_{k=1}^{q}\hat{x}_k$.

The next theorem states the main concentration result for the single sketch estimator. Both upper and lower bounds are given for the ratio of the cost of the sketched solution to the optimal solution. The full proof is given in the Appendix.

Theorem 2.4 (Concentration bound for the single sketch estimator $\hat{x}$): Suppose that the sketch size is such that $m \gtrsim d$. Then, the optimality ratio of $\hat{x}$ with respect to the optimal solution x^* is concentrated around its mean as

$$P\left(\left|\frac{f(\hat{x})}{f(x^*)} - 1 - \frac{d}{m-d-1}\right| < \epsilon\right) \geq 1 - C_1 e^{-C_2\epsilon^4 m} \tag{9}$$

where C_1, C_2 are positive constants.

Proof sketch: Our proof technique involves the concentration of Gaussian quadratic forms for Sb conditioned on SA. We leverage that the error $\|A(\hat{x} - x^*)\|_2^2$ is a Gaussian quadratic when conditioned on SA as shown in (7). In particular, we use the concentration result given in Lemma 2.5 for quadratic form of Gaussian random variables.

Lemma 2.5 (Concentration of Gaussian quadratic forms, [32]): Let the entries of $z \in \mathbb{R}^m$ be distributed as i.i.d. $\mathcal{N}(0,1)$. For any $G \in \mathbb{R}^{m\times m}$ and $\epsilon > 0$,

$$P\big(z^TGz - \mathbb{E}[z^TGz] > 2\|G\|_F\sqrt{\epsilon} + 2\|G\|_2\epsilon\big) \leq e^{-\epsilon}\,. \tag{10}$$

Then we note that after applying Lemma 2.5, the conditional expectation of the error is given by $\mathbb{E}[\|A(\hat{x} - x^*)\|_2^2\,|SA] = \frac{f(x^*)}{m}\operatorname{tr}((U^TS^TSU)^{-1})$. Next, we focus on the trace term $\operatorname{tr}((U^TS^TSU)^{-1})$ and relate it to the Stieltjes transform [33].

Definition 2.6 (Stieltjes Transform): We define the Stieltjes Transform for a random rectangular matrix $S \in \mathbb{R}^{m\times d}$ such that $d \leq m$ as

$$m_S(z) := \frac{1}{d}\operatorname{tr}(S^TS - zI)^{-1} \tag{11}$$

$$= \frac{1}{d}\sum_{i=1}^{d}\frac{1}{\lambda_i(S^TS) - z}\,. \tag{12}$$

Here $\lambda_i(S^TS)$ denotes the i'th eigenvalue of the symmetric matrix S^TS.

We derive a high probability bound for the trace term around its expectation by leveraging the concentration of the empirical Stieltjes transform to its expectation, which is given in Lemma 2.7 below.

Lemma 2.7: The trace $\operatorname{tr}((U^TS^TSU)^{-1})$ is concentrated around its mean with high probability as follows

$$P\Big(|\operatorname{tr}((U^TS^TSU)^{-1}) - \mathbb{E}[\operatorname{tr}((U^TS^TSU)^{-1})]| \leq \epsilon\Big) \geq 1 - 4e^{-\frac{\epsilon^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} - e^{-m\delta^2/2}\,, \tag{13}$$

for any $\epsilon, \delta > 0$.

The proof of Lemma 2.7 is provided in the Appendix. Combining the concentration of the Stieltjes transform with the concentration of Gaussian quadratic forms completes the proof of Theorem 2.4.

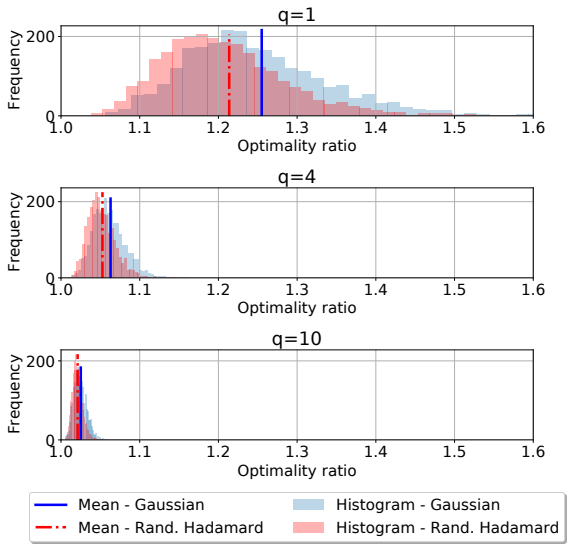


Fig. 2. Histograms of the optimality ratio $f(\bar{x})/f(x^*)$ for both Gaussian sketch and randomized Hadamard sketch. The vertical lines indicate the empirical mean of the error. The dataset is synthetically generated with $n = 512, d = 20$ and the sketch size is $m = 100$. The histogram is calculated over 2500 independent trials. The plot at the top shows the performance of the single sketch estimator, and the middle and bottom plots are for the averaged estimator with $q = 4$ and $q = 10$ workers, respectively.

Next, we show that the relative error of the averaged estimator is also concentrated with exponentially small failure probability.

Theorem 2.8 (Concentration bound for the averaged estimator $\bar{x}$): Let the sketch size satisfy $m \gtrsim d$. Then, the ratio of the cost for the averaged estimator $\bar{x} = \frac{1}{q}\sum_{k=1}^{q} \hat{x}_k$ to the optimal cost is concentrated around its mean as follows

$$P\left(\left|\frac{f(\bar{x})}{f(x^*)} - 1 - \frac{1}{q}\frac{d}{m-d-1}\right| < \epsilon\right) \geq 1 - qC_1 e^{-C_2(q\epsilon)^4 m} \tag{14}$$

where C_1, C_2 are positive constants.

The above result shows that the relative error of the averaged estimator using distributed sketching concentrates considerably faster compared to a single sketch. Moreover, the above bound offers a method to choose the values of the sketch size m and number of workers q to achieve a desired relative error with exponentially small error probability. This concentration result provides more insight when ϵ is not smaller than $O(1/m^{1/4})$, e.g., when ϵ is a constant. We note that it may be possible to improve the ϵ^4 dependence in future work.

Figure 2 shows the histograms for the single sketch ($q = 1$) and averaged estimator ($q = 4$ and $q = 10$) for Gaussian and randomized Hadamard sketches. This experiment demonstrates that the optimality ratio $f(\bar{x})/f(x^*)$ is concentrated around its mean and its mean and concentration improve as we increase the number of workers q.

C. Error Lower Bounds via Fisher Information

We now present an error lower bound result for the Gaussian sketch. We first consider single sketch estimators and then discuss the distributed sketching case. Two different lower bounds can be obtained depending on whether the estimator is restricted to be unbiased or not. Lemma 2.9 and 2.10 provide error lower bounds for all unbiased and general (i.e., possibly biased) estimators, respectively. The proof of Lemma 2.9 is based on Fisher information and Cramér-Rao lower bound while Lemma 2.10 additionally employs the van Trees inequality. We refer the reader to [31] for details on these single sketch lower bounds.

Lemma 2.9 (Unbiased estimators, [31]): For any single sketch *unbiased* estimator $\hat{x}$ obtained from the Gaussian sketched data SA and Sb, the expected error is lower bounded as follows

$$\mathbb{E}[f(\hat{x})] - f(x^*) \geq f(x^*)\frac{d}{m-d-1}. \tag{15}$$

We note that the expected error of the single sketch estimator that we provide in Lemma 2.1 exactly matches the error lower bound in Lemma 2.9. Therefore, we conclude that no other unbiased estimator based on a single sketch SA, Sb can achieve a better expected relative error.

Lemma 2.10 (General estimators, [31]): For any single sketch estimator $\hat{x}$, which is possibly biased, obtained from the Gaussian sketched data SA and Sb, the expected error is lower bounded as follows

$$\mathbb{E}[f(\hat{x})] - f(x^*) \geq f(x^*)\frac{d}{m}. \tag{16}$$

We now provide a novel generalized lower bound that applies to averaged estimators $\bar{x}$ obtained via distributed sketching. Theorem 2.11 states the main lower bound result for the averaged estimator.

Theorem 2.11: For any averaged estimator $\bar{x} = \frac{1}{q}\sum_{k=1}^{q} \hat{x}_k$, where each $\hat{x}_k$ is based on Gaussian sketched data S_kA, S_kb, $k = 1, \ldots, q$, the expected error is lower bounded as follows

(i) for unbiased estimators satisfying $\mathbb{E}[\hat{x}_k] = x^*$

$$\mathbb{E}[f(\bar{x})] - f(x^*) \geq \frac{f(x^*)}{q}\frac{d}{m-d-1}. \tag{17}$$

(ii) for general (i.e., possibly biased) estimators

$$\mathbb{E}[f(\bar{x})] - f(x^*) \geq \frac{f(x^*)}{q}\frac{d}{m}. \tag{18}$$

We note that the expected error of the averaged estimator given in Theorem 2.2 matches exactly the lower bound in Theorem 2.11 for unbiased estimators. For general estimators, we observe that the lower bound matches the upper bound for large m, e.g., when $m - d - 1 = O(m)$.

D. Distributed Sketching for Least-Norm Problems

In this section, we consider the underdetermined case where $n < d$ and applying the sketching matrix from the right, i.e., on the features. We will refer to this method as *right sketch*. Let us define the minimum norm solution

$$x^* = \arg\min_x \|x\|_2^2 \quad \text{s.t. } Ax = b. \tag{19}$$

The above problem has a closed-form solution given by $x^* = A^T(AA^T)^{-1}b$ when the matrix A is full row rank. We will assume that the full row rank condition holds in the sequel. Let

us denote the optimal value of the minimum norm objective as $f(x^*) = \|x^*\|_2^2 = b^T(AA^T)^{-1}b$. The k'th worker node will compute the approximate solution

$$\hat{z}_k = \arg\min_z \|z\|_2^2 \quad \text{s.t. } AS_k^T z = b, \tag{20}$$

where $S_k \in \mathbb{R}^{m\times d}$ and $z \in \mathbb{R}^m$. Then, the estimate $\hat{x}_k$ will be computed using $\hat{x}_k = S_k^T\hat{z}_k$ which is followed by the communication of the estimate $\hat{x}_k$ to the master node. The averaged solution is computed at the master node as $\bar{x} = \frac{1}{q}\sum_{k=1}^q \hat{x}_k$. We will assume that the sketch matrices S_k are i.i.d. Gaussian in deriving the error expressions. Lemma 2.12 establishes the approximation error for a single right sketch estimator.

Lemma 2.12: For the Gaussian sketch with sketch size $m > n+1$, the estimator $\hat{x}$ satisfies

$$\mathbb{E}[\|\hat{x} - x^*\|_2^2] = \frac{d-n}{m-n-1} f(x^*).$$

Proof of Lemma 2.12: Conditioned on AS^T, we have

$$\hat{x} \sim \mathcal{N}\Big(x^*, P_{\text{Null}(A)}\|AS^T(AS^TSA^T)^{-1}b\|_2^2\Big).$$

Noting that $\mathbb{E}[(AS^TSA^T)^{-1}] = AA^T\frac{m}{m-n-1}$, taking the expectation and noting that $\text{tr}(P_{\text{Null}(A)}) = d-n$, we obtain

$$\mathbb{E}[\|\hat{x} - x^*\|_2^2] = \frac{d-n}{m-n-1} b^T(AA^T)^{-1}b = \frac{d-n}{m-n-1} f(x^*). \tag{21}$$

■

An exact formula for averaging multiple outputs in right sketch that parallels Theorem 2.2 can be obtained in a similar fashion:

$$\mathbb{E}[\|\bar{x} - x^*\|_2^2] = \frac{1}{q}\,\mathbb{E}[\|\hat{x}_k - x^*\|_2^2] = \frac{1}{q}\frac{d-n}{m-n-1} f(x^*).$$

Hence, for the distributed Gaussian right sketch, we establish the approximation error as stated in Theorem 2.13.

Theorem 2.13 (Cost approximation for least-norm problems): Let S_k, $k = 1,\dots,q$ be Gaussian sketching matrices, then the error of the averaged solution $\bar{x}$ satisfies

$$\frac{\mathbb{E}[f(\bar{x})] - f(x^*)}{f(x^*)} = \frac{1}{q}\frac{d-n}{m-n-1}. \tag{22}$$

Consequently, Markov's inequality implies that $\frac{f(\bar{x})-f(x^*)}{f(x^*)} \le \epsilon$ holds with probability at least $\left(1 - \frac{1}{q\epsilon}\frac{d-n}{m-n-1}\right)$ for any $\epsilon > 0$.

III. Distributed Sketching for Iterative Algorithms

A. Distributed Iterative Hessian Sketch

In this section, we consider an iterative algorithm for solving the unregularized least squares problem in (4), where $\lambda_1 = 0$ with higher accuracy. Let us consider applying Newton's method to this problem

$$x_{t+1} = x_t - \mu(A^TA)^{-1}A^T(Ax_t - b), \tag{23}$$

where $\mu > 0$ is the step size. Note that, Newton's method terminates in one step since the Hessian is A^TA and the update reduces to directly solving the normal equations $(A^TA)x = A^Tb$ for $\mu = 1$. However, the computational cost of this direct solution is often prohibitive for large scale problems. To remedy this, the method of iterative Hessian sketch introduced in [3] employs a randomly sketched Hessian $A^TS_t^TS_t^TA$ as follows

$$x_{t+1} = x_t - \mu(A^TS_t^TS_tA)^{-1}A^T(Ax_t - b),$$

where S_t corresponds to the sketching matrix at iteration t. Sketching reduces the row dimension of the data from n to m and hence computing an approximate Hessian $A^TS_t^TS_tA$ is computationally cheaper than the exact Hessian A^TA. Moreover, for regularized problems one can choose m to be smaller than d as we investigate in Section IV-A.

Algorithm 2: Distributed Iterative Hessian Sketch

Input: Number of iterations T, step size μ.
for $t = 1$ to T **do**
  for worker $k = 1,\dots,q$ in parallel **do**
    Obtain the sketched data $S_{t,k}A$.
    Compute gradient $g_t = A^T(Ax_t - b)$.
    Solve $\hat{\Delta}_{t,k} = \arg\min_\Delta \frac{1}{2}\|S_{t,k}A\Delta\|_2^2 + g_t^T\Delta$ and send to master node.
  end for
  Master node: Update $x_{t+1} = x_t + \mu\frac{1}{q}\sum_{k=1}^q \hat{\Delta}_{t,k}$ and send x_{t+1} to worker nodes.
end for
return x_T

In a distributed computing setting, one can obtain more accurate update directions by averaging multiple trials, where each worker node computes an independent estimate of the update direction. These approximate update directions can be averaged at the master node and the following update takes place

$$x_{t+1} = x_t - \mu\frac{1}{q}\sum_{k=1}^q (A^TS_{t,k}^TS_{t,k}A)^{-1}A^T(Ax_t - b). \tag{24}$$

Here $S_{t,k}$ is the sketching matrix for the k'th node at iteration t. The details of the distributed IHS algorithm are given in Algorithm 2. We note that although the update equation involves a matrix inverse term, in practice, this can be replaced with an approximate linear system solver. In particular, it might be computationally more efficient for worker nodes to compute their approximate update directions using indirect methods such as conjugate gradient.

We note that in Algorithm 2, worker nodes communicate their approximate update directions and not the approximate Hessian matrix, which reduces the communication complexity from $O(d^2)$ to $O(d)$ for each worker per iteration.

We establish the convergence rate for Gaussian sketch in Theorem 3.2, which provides an exact characterization of the expected error. First, we give the definition of the error in Definition 3.1.

Definition 3.1: To quantify the approximation quality of the iterate $x_t \in \mathbb{R}^d$ with respect to the optimal solution $x^* \in \mathbb{R}^d$, we define the error as $e_t^A \coloneqq A(x_t - x^*)$.

To state the result, we first introduce the following moments of the inverse Wishart distribution (see the appendix).

$$\theta_1 \coloneqq \frac{m}{m-d-1},$$
$$\theta_2 \coloneqq \frac{m^2(m-1)}{(m-d)(m-d-1)(m-d-3)}. \quad (25)$$

Theorem 3.2 (Expected error decay for Gaussian sketch): In Algorithm 2, let us set $\mu = 1/\theta_1$ and assume $S_{t,k}$'s are i.i.d. Gaussian sketches, then the expected squared norm of the error e_t^A evolves according to the following relation:

$$\mathbb{E}[\|e_{t+1}^A\|_2^2] = \frac{1}{q}\left(\frac{\theta_2}{\theta_1^2} - 1\right)\|e_t^A\|_2^2.$$

Next, Corollary 3.3 builds on Theorem 3.2 to characterize the number of iterations for Algorithm 2 to achieve an error of ϵ. The number of iterations required for error ϵ scales with $\log(1/\epsilon)/\log(q)$.

Corollary 3.3: Let $S_{t,k} \in \mathbb{R}^{m\times n}$ $(t = 1,\ldots,T,\ k = 1,\ldots,q)$ be Gaussian sketching matrices. Then, Algorithm 2 outputs x_T that is ϵ-accurate with respect to the initial error in expectation, that is, $\frac{\mathbb{E}[\|e_T^A\|_2^2}{\|Ax^*\|_2^2} = \epsilon$ where T is given by

$$T = \frac{\log(1/\epsilon)}{\log(q) - \log\left(\frac{\theta_2}{\theta_1^2} - 1\right)},$$

where the overall required communication is Tqd real numbers, and the computational complexity per worker is

$$O(Tmnd + Tmd^2 + Td^3).$$

Remark 3.4: Provided that m is at least $2d$, the term $\log\left(\frac{\theta_2}{\theta_1^2} - 1\right)$ is negative. Hence, T is upper-bounded by $\frac{\log(1/\epsilon)}{\log(q)}$.

Distributed iterative Hessian sketch (IHS) and its convergence analysis have been considered for the first time in this work. Our technique leads to exact expected error expressions for Gaussian sketches. A particular way that the expected error result can be viewed is through the lens of massively parallel computing (where q is very large). In this case, as q gets larger, the error will converge to the expected error. [34] presents high probability bounds for the error of the distributed IHS algorithm, which we proposed and studied in this work. Differently from our work, the result of [34] assumes surrogate sketches. Surrogate sketches are defined by modifying standard sketching techniques using determinantal point processes. [34] also extends the result on distributed IHS to distributed Newton sketch.

B. Distributed Newton Sketch

The update equation for Newton's method is of the form $x_{t+1} = x_t - \alpha_1 H_t^{-1} g_t$, where $H_t \in \mathbb{R}^{d\times d}$ and $g_t \in \mathbb{R}^d$ denote the Hessian matrix and the gradient vector at iteration t respectively, and α_1 is the step size. In contrast, Newton Sketch performs the approximate updates

$$x_{t+1} = x_t + \alpha_1 \arg\min_{\Delta}\left(\frac{1}{2}\|S_t H_t^{1/2}\Delta\|_2^2 + g_t^T\Delta\right), \quad (26)$$

Algorithm 3: Distributed Newton Sketch

Input: Tolerance ϵ
while $g_t^T\left(\sum_{k=1}^q \hat{\Delta}_{t,k}\right)/2 \le \epsilon$ **do**
  for worker $k = 1,\ldots,q$ in parallel **do**
    Obtain $S_{t,k}H_t^{1/2}$.
    Obtain the gradient g_t.
    Compute approximate Newton direction $\hat{\Delta}_{t,k} = \arg\min_\Delta(\frac{1}{2}\|S_{t,k}H_t^{1/2}\Delta\|_2^2 + g_t^T\Delta)$ and send to master node.
  end for
  Master node: Determine step size α_2 and update $x_{t+1} = x_t + \alpha_2\frac{1}{q}\sum_{k=1}^q \hat{\Delta}_{t,k}$.
end while

where the sketching matrices $S_t \in \mathbb{R}^{m\times n}$ are refreshed every iteration. There can be a multitude of options for devising a *distributed* Newton's method or a *distributed* Newton sketch algorithm. Here we consider a scheme that is similar in spirit to the GIANT algorithm [23] where worker nodes communicate length-d approximate update directions to be averaged at the master node. Another alternative scheme would be to communicate the approximate Hessian matrices, which would require an increased communication load of d^2 numbers.

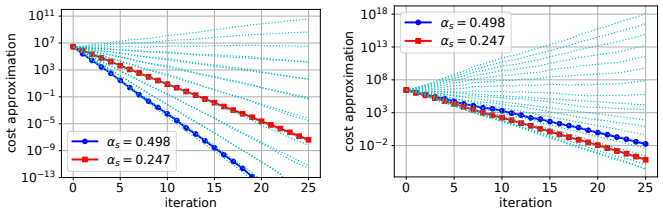
We consider Hessian matrices of the form $H_t = (H_t^{1/2})^T H_t^{1/2}$, where we assume that $H_t^{1/2} \in \mathbb{R}^{n\times d}$ is a full column rank matrix and $n \ge d$. Note that this factorization is already available in terms of scaled data matrices in various problems. For instance, in the Least Squares, we simply have $H_t^{1/2} = A$. More generally, for functions of type $f(Ax)$, the Hessian takes the form $A^T f''(Ax)A$. The factorization $H_t = (H_t^{1/2})^T H_t^{1/2}$ can be computed by simply considering the factorization of $f''(Ax)$. For large scale data matrices A, this factorization could be computationally inexpensive as long as $f''(Ax)$ is easy to factorize, e.g., diagonal $f''(Ax)$. Please refer to Section VI for further details and examples including logistic regression and inequality constrained optimization.

The factorization of the Hessian matrix enables the fast construction of an approximation of H_t by applying sketching $S_t H_t^{1/2}$ which leads to the approximation $\hat{H}_t = (S_t H_t^{1/2})^T S_t H_t^{1/2}$. Averaging for regularized problems with Hessian matrices of the form $H_t = (H_t^{1/2})^T H_t^{1/2} + \lambda_1 I_d$ will be considered in the next subsection.

The update equation for *distributed* Newton sketch for a system with q worker nodes can be written as

$$x_{t+1} = x_t + \alpha_2\frac{1}{q}\sum_{k=1}^q \arg\min_\Delta\left(\frac{1}{2}\|S_{t,k}H_t^{1/2}\Delta\|_2^2 + g_t^T\Delta\right). \quad (27)$$

Note that the above update requires access to the full gradient g_t. If worker nodes do not have access to the entire dataset, then this requires an additional communication round per iteration where worker nodes communicate their local gradients with the master node, which computes the full gradient and broadcasts to worker nodes. The details of the distributed Newton sketch method are given in Algorithm 3.



(a) $q = 10$ workers (b) $q = 2$ workers

Fig. 3. Cost approximation $(f(x_t) - f(x^*))/f(x^*)$ for Algorithm 3 against iteration number t for various step sizes in solving a linear least squares problem on randomly generated data. The cyan colored dotted lines show cost approximation when we make a search for the learning rate α_s between 0.05 and 1. The blue line with circle markers corresponds to $\alpha_s = 1/\theta_1$ that leads to the unbiased estimator and the red line with square markers corresponds to $\alpha_s = \theta_1/\theta_2$ that gives the minimum variance. The resulting step size scaling factors α_s are shown in the legends of the plots. The parameters used in these experiments are $n = 1000$, $d = 200$, $m = 400$.

We analyze the bias and the variance of the update directions for distributed Newton sketch, and derive exact expressions for Gaussian sketching matrices. First, we establish the notation for update directions. We will let Δ_t^* denote the optimal Newton update direction at iteration t

$$\Delta_t^* = ((H_t^{1/2})^T H_t^{1/2})^{-1} g_t \tag{28}$$

and let $\hat{\Delta}_{t,k}$ denote the approximate update direction returned by worker node k at iteration t, which has the closed form expression

$$\hat{\Delta}_{t,k} = \alpha_s \left((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2} \right)^{-1} g_t. \tag{29}$$

Note that the step size for the averaged update direction will be calculated as $\alpha_2 = \alpha_1 \alpha_s$. Theorem 3.5 characterizes how the update directions need to be scaled to obtain an unbiased update direction, and also a minimum variance estimator.

Theorem 3.5: For Gaussian sketches $S_{t,k}$, assuming $H_t^{1/2}$ is full column rank, the variance $\mathbb{E}[\|H_t^{1/2}(\hat{\Delta}_{t,k} - \Delta_t^*)\|_2^2]$ is minimized when α_s is chosen as $\alpha_s = \frac{\theta_1}{\theta_2}$ whereas the bias $\mathbb{E}[H_t^{1/2}(\hat{\Delta}_{t,k} - \Delta_t^*)]$ is zero when $\alpha_s = \frac{1}{\theta_1}$, where θ_1 and θ_2 are as defined in (25).

Figure 3 demonstrates that choosing $\alpha_2 = \alpha_1 \alpha_s$ when α_s is calculated using the unbiased estimator formula $\alpha_s = 1/\theta_1$ leads to faster decrease of the objective value when the number of workers is high. If the number of workers is small, one should choose the step size that minimizes variance using $\alpha_s = \theta_1/\theta_2$ instead. Furthermore, Figure 3(a) illustrates that the blue curve with square markers is in fact the best one could hope to achieve as it is very close to the best cyan dotted line.

We note that our bias correction result considers Gaussian sketch. In [34], our bias correction result is extended to surrogate sketches. Surrogate sketches typically allow exact expectation expressions to be obtained. The theoretical analysis for the Gaussian case requires identical singular values for the data matrix in deriving exact bias expressions. This assumption is relaxed in [34] for surrogate sketches.

IV. Bias Correction for Regularized Problems

A. Bias Correction for Regularized Least Squares

We have so far studied distributed randomized regression for unregularized problems and showed that Gaussian sketch leads to unbiased estimators. In this section, we focus on the regularized case and show that using the original regularization coefficient for the sketched problems causes the estimators to be biased. In addition, we provide a bias correction procedure. More precisely, the method described in this section is based on non-iterative averaging for solving the linear least squares problem with ℓ_2 regularization, i.e., ridge regression.

Note that we have λ_1 as the regularization coefficient of the original problem and λ_2 for the sketched sub-problems. If λ_2 is chosen to be equal to λ_1, then this scheme reduces to the framework given in the work of [2] and we show in Theorem 4.3 that $\lambda_2 = \lambda_1$ leads to a biased estimator, which does not converge to the optimal solution.

We first introduce the following results on traces involving random Gaussian matrices which are instrumental in our result.

Lemma 4.1 ([35]): For a Gaussian sketching matrix S, the following asymptotic formula holds

$$\lim_{n\to\infty} \mathbb{E}[\mathrm{tr}((U^T S^T S U + \lambda_2 I)^{-1})] = d \times \theta_3(d/m, \lambda_2),$$

where $\theta_3(d/m, \lambda_2) :=$

$$= \frac{-\lambda_2 + d/m - 1 + \sqrt{(-\lambda_2 + d/m - 1)^2 + 4\lambda_2 d/m}}{2\lambda_2 d/m}.$$

Lemma 4.2: For a Gaussian sketching matrix S, the following asymptotic formula holds

$$\lim_{n\to\infty} \mathbb{E}[(U^T S^T S U + \lambda_2 I)^{-1}] = \theta_3(d/m, \lambda_2) I_d,$$

where $\theta_3(d/m, \lambda_2)$ is as defined in Lemma 4.1.

We list the distributed randomized ridge regression method in Algorithm 4. This algorithm assumes that our goal is to solve a large scale regression problem for a given value of regularization coefficient λ_1. Theorem 4.3 states the main result of this subsection. In short, for the averaged result to converge to the optimal solution x^*, the regularization coefficient needs to be modified to λ_2^*.

Theorem 4.3: Given the thin SVD decomposition $A = U\Sigma V^T \in \mathbb{R}^{n\times d}$ and $n \geq d$, and assuming A has full rank and has identical singular values (i.e., $\Sigma = \sigma I_d$ for some $\sigma > 0$), there is a value of λ_2 that yields a zero bias of the single sketch estimator $\mathbb{E}[A(\hat{x}_k - x^*)]$ as n tends to infinity if

(i) $m > d$ or
(ii) $m \leq d$ and $\lambda_1 \geq \sigma^2 \left(\frac{d}{m} - 1\right)$

and the value of λ_2 that achieves zero bias is given by

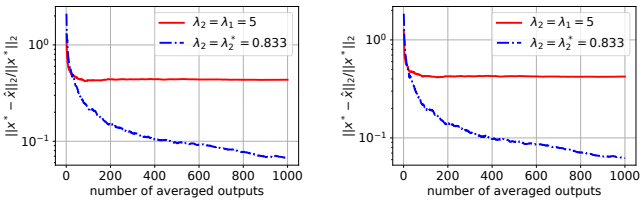
$$\lambda_2^* = \lambda_1 - \frac{d}{m} \frac{\lambda_1}{1 + \lambda_1/\sigma^2}, \tag{30}$$

where the matrix S_k in $\hat{x}_k = \arg\min_x \|S_k A x - S_k b\|_2^2 + \lambda_2 \|x\|_2^2$ is the Gaussian sketch.

Figure 4 illustrates the practical implications of Theorem 4.3. If λ_2 is chosen according to the formula in (30), then the averaged solution $\bar{x}$ gives a significantly better approximation to x^* than if we had used $\lambda_2 = \lambda_1$. The data matrix A in

Algorithm 4: Distributed Randomized Ridge Regression

Set σ to the mean of singular values of A.
Calculate $\lambda_2^* = \lambda_1 - \frac{d}{m}\frac{\lambda_1}{1+\lambda_1/\sigma^2}$.
for worker $k = 1, \ldots, q$ in parallel **do**
  Obtain the sketched data and sketched output: $S_k A$ and $S_k b$.
  Solve $\hat{x}_k = \arg\min_x \|S_k A x - S_k b\|_2^2 + \lambda_2^* \|x\|_2^2$ and send $\hat{x}_k$ to master node.
end for
Master node: return $\bar{x} = \frac{1}{q}\sum_{k=1}^{q} \hat{x}_k$.



(a) Identical singular values (b) Non-identical singular values

Fig. 4. Plots of $\|\bar{x} - x^*\|_2/\|x^*\|_2$ against the number of averaged worker outputs for an unconstrained least squares problem with regularization using Algorithm 4. The dashed blue line corresponds to the case where λ_2 is determined according to the formula (30), and the solid red line corresponds to the case where λ_2 is the same as λ_1. The parameters are as follows: $n = 1000$, $d = 100$, $\lambda_1 = 5$, $m = 20$, sketch type is Gaussian. (a) All singular values of A are 1. (b) Singular values of A are not identical and their mean is 1.

Figure 4(a) has identical singular values, and 4(b) shows the case where the singular values of A are not identical. When the singular values of A are not all equal to each other, we set σ to the mean of the singular values of A as a heuristic, which works extremely well as shown in Figure 4(b). According to the formula in (30), the value of λ_2 that we need to use to achieve zero bias is found to be $\lambda_2^* = 0.833$ whereas $\lambda_1 = 5$. The plot in Figure 4(b) illustrates that even if the assumption that $\Sigma = \sigma I_d$ in Theorem 4.3 is violated, the proposed bias corrected averaging method outperforms vanilla averaging in [2] where $\lambda_2 = \lambda_1$.

Remark 4.4 (Varying sketch sizes): Let us now consider the scenario where we have different sketch sizes for each worker node. This situation frequently arises in heterogeneous computing environments. Specifically, let us assume that the sketch size for worker k is m_k, $k = 1, \ldots, q$. It follows from Theorem 4.3 that by choosing the regularization parameter for worker node k as

$$\lambda_2^*(k) = \lambda_1 - \frac{d}{m_k}\frac{\lambda_1}{1+\lambda_1/\sigma^2},$$

it is possible to obtain unbiased estimators $\hat{x}_k$, $k = 1, \ldots, q$ and hence an unbiased averaged result $\bar{x}$. Note that here we assume that the sketch size for each worker satisfies the condition in Theorem 4.3, that is, either $m_k > d$ or $m_k \leq d$ and $\lambda_1 \geq \sigma^2(d/m_k - 1)$.

B. Distributed Newton Sketch for Regularized Problems

We now consider problems with squared ℓ_2-norm regularization. In particular, we study problems with Hessian matrices of the form $H_t = (H_t^{1/2})^T H_t^{1/2} + \lambda_1 I_d$. Sketching can be applied to obtain approximate Hessian matrices as $H_t = (S_t H_t^{1/2})^T S_t H_t^{1/2} + \lambda_2 I_d$. Note that the case $\lambda_2 = \lambda_1$ corresponds to the setting in the GIANT algorithm described in [23] when the sketch is uniform row sampling. The theoretical result given in this section assumes Gaussian sketch. As we will show in the numerical results section, using the bias correction formula improves the performance of other sketch types as well in our experiments.

Theorem 4.5 establishes that λ_2 should be chosen according to the formula (31) under the assumption that the singular values of $H_t^{1/2}$ are identical. We later verify empirically that when the singular values are not identical, plugging the mean of the singular values into the formula still leads to improvements over the case of $\lambda_2 = \lambda_1$.

Theorem 4.5: Given the thin SVD decomposition $H_t^{1/2} = U\Sigma V^T \in \mathbb{R}^{n\times d}$ and $n \geq d$ where $H_t^{1/2}$ is assumed to have full rank and satisfy $\Sigma = \sigma I_d$, the bias of the single sketch Newton step estimator $\mathbb{E}[H_t^{1/2}(\hat{\Delta}_{t,k} - \Delta_t^*)]$ is equal to zero as n goes to infinity when λ_2 is chosen as

$$\lambda_2^* = \left(\lambda_1 + \frac{d}{m}\sigma^2\right)\left(1 - \frac{d/m}{1 + \lambda_1\sigma^{-2} + d/m}\right), \qquad (31)$$

where $\Delta_t^* = ((H_t^{1/2})^T H_t^{1/2} + \lambda_1 I_d)^{-1} g_t$ and $\hat{\Delta}_{t,k} = ((S_{t,k} H_t^{1/2})^T S_{t,k} H_t^{1/2} + \lambda_2 I_d)^{-1} g_t$, and $S_{t,k}$ is the Gaussian sketch.

Remark 4.6: The proof of Theorem 4.5 builds on the proof of Theorem 4.3 (see the appendix). The main difference between these results is that the setting in Theorem 4.5 is such that we assume access to the full gradient, i.e. only the Hessian matrix is sketched. In contrast, the setting in Theorem 4.3 does not use the exact gradient.

V. Privacy Preserving Properties

We now digress from the error and convergence properties of distributed sketching methods to consider the privacy preserving properties of distributed sketching. We use the notion of differential privacy for our privacy result given in Definition 5.1. Differential privacy is a worst case type of privacy notion which does not require distribution assumptions on the data. It has been the privacy framework adopted by many works in the literature.

Definition 5.1 ((ε, δ)-Differential Privacy, [36], [37]): An algorithm ALG which maps $(n \times d)$-matrices into some range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for all pairs of neighboring inputs A and A' (i.e. they differ only in a single row) and all subsets $\mathcal{S} \subset \mathcal{R}$, it holds that $P(\text{ALG}(A) \in \mathcal{S}) \leq e^{\varepsilon} P(\text{ALG}(A') \in \mathcal{S}) + \delta$.

Theorem 5.2 characterizes the required conditions and the sketch size to guarantee (ε, δ)-differential privacy for a given dataset. The proof of Theorem 5.2 is based on the (ε, δ)-differential privacy result for i.i.d. Gaussian random projections from [37]. Note that Theorem 5.2 assumes a sketch size

required for privacy. The sketch size depends on whether we consider privacy against all q worker nodes or a single node.

Theorem 5.2: Given the data matrix $A \in \mathbb{R}^{n\times d}$ and the output vector $b \in \mathbb{R}^n$, let $A_c := [A, b] \in \mathbb{R}^{n\times(d+1)}$. Define $\beta := \ln(4/\delta)$. Define $\sigma_0 := \sigma_{\min}(A_c)/\sqrt{n}$, and $B_0 := \max_{i,j} |A_{c,ij}|$. Suppose that the conditions

$$\frac{n}{d+1} \geq \left(3 + \frac{2\beta}{\varepsilon}\right)\frac{B_0^2}{\sigma_0^2} \qquad \text{and} \qquad m > d+1 \tag{32}$$

are satisfied and the sketch size satisfies

$$m = O\left(\beta \frac{n^2}{(d+1)^2}\frac{\varepsilon^2}{(\varepsilon+\beta)^2}\right) \text{ for privacy w.r.t. one node, or} \tag{33}$$

$$m = O\left(\frac{\beta}{q} \frac{n^2}{(d+1)^2}\frac{\varepsilon^2}{(\varepsilon+\beta)^2}\right) \text{ for privacy w.r.t. } q \text{ nodes.} \tag{34}$$

Then, for Gaussian sketch matrices $S_k \in \mathbb{R}^{m\times n}$, publishing the sketched matrices $S_k A_c \in \mathbb{R}^{m\times(d+1)}$, $k = 1, \ldots, q$ is (ε, δ)-differentially private for any $\varepsilon > 0$, $\beta > 1 + \ln(4)$.

Remark 5.3: For fixed values of β, $\sigma_{\min}$, B_0, n and d, the approximation error is on the order of $O\left(\frac{1}{\varepsilon^2}\right)$ for (ε, δ)-differential privacy with respect to all q workers.

The work [38] considers convex optimization under privacy constraints and shows that ε-differential privacy (i.e. equivalent to Definition 5.1 with $\delta = 0$), the approximation error of their distributed iterative algorithm is on the order of $O(\frac{1}{\varepsilon^2})$, which is on the same order as Algorithm 1. The two algorithms however have different dependencies on parameters, which are hidden in the O-notation. We note that the approximation error of the algorithm in [38] depends on parameters that Algorithm 1 does not have such as the initial step size, the step size decay rate, and noise decay rate. The reason for this is that the algorithm in [38] is a synchronous iterative algorithm designed to solve a more general class of optimization problems. Algorithm 1, on the other hand, is designed to solve linear regression problems and offers a significant advantage due to its single-round communication requirement.

Remark 5.4: A similar privacy guarantee holds for the right sketch method discussed in Section II-D. In right sketch, we only sketch the data matrix A and not the output vector b. For publishing AS_k^T to be (ε, δ)-differentially private, Theorem 5.2 still holds with the modification that we replace A_c with A^T.

VI. Applications of Distributed Sketching

This section describes some example problems where our methodology can be applied. In particular, the problems in this section are convex problems that are efficiently addressed by our methods in distributed systems. We present numerical results on these problems in Section VII.

A. Logistic Regression

We begin by considering the well-known logistic regression model with squared ℓ_2-norm penalty. The optimization problem for this model can be formulated as $\text{minimize}_x f(x)$ where

$$f(x) = -\sum_{i=1}^{n} \left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right) + \frac{\lambda_1}{2}\|x\|_2^2, \tag{35}$$

and $p \in \mathbb{R}^n$ is defined such that $p_i = 1/(1+\exp(-\tilde{a}_i^T x))$. $\tilde{a}_i \in \mathbb{R}^d$ denotes the i'th row of the data matrix $A \in \mathbb{R}^{n\times d}$. The output vector is denoted by $y \in \mathbb{R}^n$.

We can use the distributed Newton sketch algorithm to solve this optimization problem. The gradient and Hessian for $f(x)$ are as follows

$$g = A^T(p-y) + \lambda_1 x,$$
$$H = A^T D A + \lambda_1 I_d\,.$$

D is a diagonal matrix with the elements of the vector $p(1-p)$ as its diagonal entries. The sketched Hessian matrix in this case can be formed as $(SD^{1/2}A)^T(SD^{1/2}A) + \lambda_2^* I_d$ where λ_2^* can be calculated using (31), and we can set σ to the mean of the singular values of $D^{1/2}A$. Since the entries of the matrix D are a function of the variable x, the matrix D gets updated every iteration. It might be computationally expensive to recompute the mean of the singular values of $D^{1/2}A$ in every iteration. However, we have found through experiments that it is not required to compute the exact value of the mean of the singular values for bias reduction. For instance, setting σ to the mean of the diagonals of the matrix $D^{1/2}$ as a heuristic works sufficiently well.

B. Inequality Constrained Optimization

The second application that we consider is the inequality constrained optimization problem of the form

$$\begin{aligned} \text{minimize}_x \quad & \|x - c\|_2^2 \\ \text{subject to} \quad & \|Ax\|_\infty \leq \lambda \end{aligned} \tag{36}$$

where $A \in \mathbb{R}^{n\times d}$, and $c \in \mathbb{R}^d$ are the problem data, and $\lambda \in \mathbb{R}$ is a positive scalar. Note that this problem is the dual of the Lasso problem given by $\min_x \lambda\|x\|_1 + \frac{1}{2}\|Ax - c\|_2^2$.

The above problem can be tackled by the standard log-barrier method [39], by solving sequences of unconstrained barrier penalized problems as follows

$$\begin{aligned} \text{minimize}_x \; & -\sum_{i=1}^{n} \log(-\tilde{a}_i^T x + \lambda) - \sum_{i=1}^{n} \log(\tilde{a}_i^T x + \lambda) \\ & + \lambda_1\|x\|_2^2 - 2\lambda_1 c^T x + \lambda_1\|c\|_2^2 \end{aligned} \tag{37}$$

where $\tilde{a}_i$ represents the i'th row of A. The distributed Newton sketch algorithm could be used to solve this problem. The gradient and Hessian of the objective are given by

$$g = -A_c^T D \mathbf{1}_{2n\times 1} + 2\lambda_1 x - 2\lambda_1 c,$$
$$H = (DA_c)^T(DA_c) + 2\lambda_1 I_d.$$

Here $A_c = [A^T, -A^T]^T$ and D is a diagonal matrix with the element-wise inverse of the vector $(A_c x - \mathbf{1}_{2n\times 1})$ as its diagonal entries. $\mathbf{1}_{2n\times 1}$ is the length-$2n$ vector of all

1's. The sketched Hessian can be written in the form of $(SDA_c)^T(SDA_c) + \lambda_2 I_d$.

Remark 6.1: Since the contribution of the regularization term in the Hessian matrix is $2\lambda_1 I_d$, we need to plug $2\lambda_1$ instead of λ_1 in the formula for computing λ_2^*.

C. Fine Tuning of Pre-Trained Neural Networks

The third application is neural network fine tuning. Let $f_{NN}(\cdot) : \mathbb{R}^d \to \mathbb{R}^r$ represent the output of the $(L-1)$'st layer of a pre-trained network which consists of L layers. We are interested in re-using a pre-trained neural network for, say, a task on a different dataset by learning a different final layer. More precisely, assuming squared loss, one way that we can formulate this problem is as follows

$$W_L^* = \arg\min_{W_L} \Bigg\| \underbrace{\begin{bmatrix} f_{NN}(\tilde{a}_1)^T \\ \vdots \\ f_{NN}(\tilde{a}_n)^T \end{bmatrix}}_{f_{NN}(A)} W_L - b \Bigg\|_F^2 + \lambda_1 \|W_L\|_F^2 \,, \tag{38}$$

where $W_L \in \mathbb{R}^{r \times C}$ is the weight matrix of the final layer. The vectors $\tilde{a}_i \in \mathbb{R}^d$ $i = 1, \ldots, n$ denote the rows of the data matrix $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^{n \times C}$ is the output matrix. In a classification problem, C would correspond to the number of classes.

For large scale datasets, it is important to develop efficient methods for solving the problem in (38). Distributed randomized ridge regression algorithm (listed in Algorithm 4) could be applied to this problem where the sketched sub-problems are of the form

$$\hat{W}_{L,k} = \arg\min_{W_L} \|S_k f_{NN}(A) W_L - S_k b\|_F^2 + \lambda_2^* \|W_L\|_F^2 \tag{39}$$

and the averaged solution can be computed at the master node as $\bar{W}_L = \frac{1}{q}\sum_{k=1}^q \hat{W}_{L,k}$. This leads to an efficient non-iterative asynchronous method for fine-tuning of a neural network.

VII. Numerical Results

We have implemented the distributed sketching methods for AWS Lambda in Python using the Pywren package [11]. The setting for the experiments is a centralized computing model where a single master node collects and averages the outputs of the q worker nodes. The majority of the figures in this section plots the approximation error which we define as $(f(\bar{x}) - f(x^*))/f(x^*)$.

A. Hybrid Sketch

In a distributed computing setting, the amount of data that can be fit into the memory of nodes and the size of the largest problem that can be solved by the nodes often do not match. The hybrid sketch idea is motivated by this mismatch and it is basically a sequentially concatenated sketching scheme where we first perform uniform sampling with dimension m' and then sketch the sampled data using another sketching method, preferably with better convergence properties (say, Gaussian) with dimension m. Worker nodes load m' rows of the data matrix A into their memory and then perform sketching, reducing the number of rows from m' to m. In addition, we note that if $m' = m$, hybrid sketch reduces to sampling and if $m' = n$, then it reduces to Gaussian sketch. The hybrid sketching scheme does not take privacy into account as worker nodes are assumed to have access to the data matrix.

For the experiments involving very large scale datasets, we have used Sparse Johnson-Lindenstrauss Transform (SJLT) [28] as the second sketching method in the hybrid sketch due to its low computational complexity.

B. Airline Dataset

We have conducted experiments with the publicly available Airline dataset [40]. This dataset contains information on domestic USA flights between the years 1987-2008. We are interested in predicting whether there is going to be a departure delay or not, based on information about the flights. More precisely, we are interested in predicting whether `DepDelay` > 15 minutes using the attributes `Month`, `DayofMonth`, `DayofWeek`, `CRSDepTime`, `CRSElapsedTime`, `Dest`, `Origin`, and `Distance`. Most of these attributes are categorical and we have used dummy coding to convert these categorical attributes into binary representations. The size of the input matrix A, after converting categorical features into binary representations, becomes $(1.21 \times 10^8) \times 774$.

We have solved the linear least squares problem on the entire airline dataset: minimize$_x \|Ax - b\|_2^2$ using q workers on AWS Lambda. The output b for the plots a and b in Figure 5 is a vector of binary variables indicating delay. The output b for the plots c and d in Figure 5 is artificially generated via $b = Ax_{truth} + \epsilon$ where x_{truth} is the underlying solution and ϵ is random Gaussian noise distributed as $\mathcal{N}(0, 0.01I)$. Figure 5 shows that sampling followed by SJLT leads to a lower error.

We note that the convergence rate gets better for higher values of m and m'. Based on the run times given in the caption of Figure 5, we see that the run times are slightly worse if SJLT is involved. Decreasing m' will help reduce this processing time at the expense of error performance.

The monetary cost incurred for this experiment on AWS Lambda is calculated as follows. Pricing for AWS Lambda (at the time of writing) is \$0.0000166667 for every GB-second. We have used $q = 100$ serverless nodes each with 3GB of memory. The longest run of the experiments in Figure 5 takes 107.6 seconds. This corresponds to a monetary cost of \$0.00538 per serverless node and \$0.538 for the entire experiment.

C. Image Dataset: Extended MNIST

The experiments of this subsection are performed on the image dataset EMNIST (extended MNIST) [41]. We have used the "bymerge" split of EMNIST, which has 700K training and 115K test images. The dimensions of the images are 28×28 and there are 47 classes in total (letters and digits). Some capital letter classes have been merged with small letter classes (like C and c), and thus there are 47 classes and not 62.

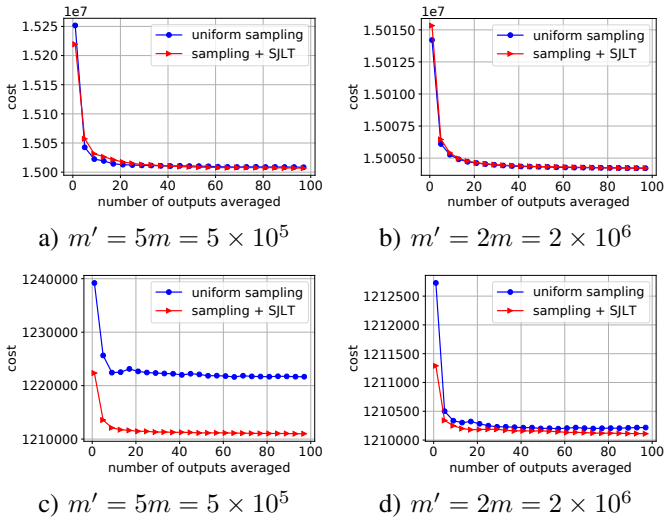


Fig. 5. AWS Lambda experiments on the entire airline dataset ($n = 1.21 \times 10^8$) with $q = 100$ workers. The run times for each plot are as follows (given in this order: sampling, sampling followed by SJLT): a: 37.5, 43.9 seconds, b: 48.3, 60.1 seconds, c: 39.8, 52.9 seconds, d: 78.8, 107.6 seconds. Here, cost refers to the training objective $\|Ax - b\|_2^2$.

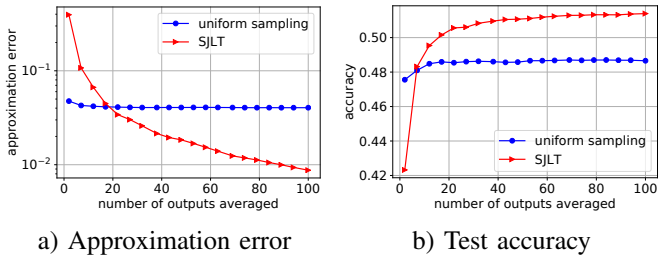


Fig. 6. Approximation error and test set classification accuracy plots for the EMNIST-bymerge dataset where $q = 100$, $m = 2000$, $s = 20$. The run times on AWS Lambda for uniform sampling and SJLT are 41.5 and 66.9 seconds, respectively.

Figure 6 shows the approximation error and test accuracy plots when we fit a least squares model on the EMNIST-bymerge dataset using the distributed randomized regression algorithm. Because this is a multi-class classification problem, we have one-hot encoded the labels. Figure 6 demonstrates that SJLT is able to drive the cost down more and leads to a better classification accuracy than uniform sampling.

D. Performance on Large Scale Synthetic Datasets

We now present the results of the experiments carried out on randomly generated large scale data to illustrate scalability of the methods. Plots in Figure 7 show the approximation error as a function of time, where the problem dimensions are as follows: $A \in \mathbb{R}^{10^7 \times 10^3}$ for plot a and $A \in \mathbb{R}^{(2\times 10^7)\times(2\times 10^3)}$ for plot b. These data matrices take up 75 GB and 298 GB, respectively. The data used in these experiments were randomly generated from the student's t-distribution with degrees of freedom of 1.5 for plot a and 1.7 for plot b. The output vector b was computed according to $b = Ax_{truth} + \epsilon$ where $\epsilon \in \mathbb{R}^n$ is i.i.d. noise distributed as $\mathcal{N}(0, 0.1I_n)$. Other

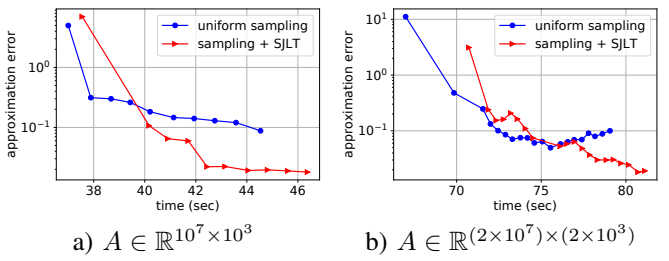


Fig. 7. Approximation error vs time for AWS Lambda experiments on randomly generated large scale datasets ($q = 200$ AWS Lambda functions have been used).

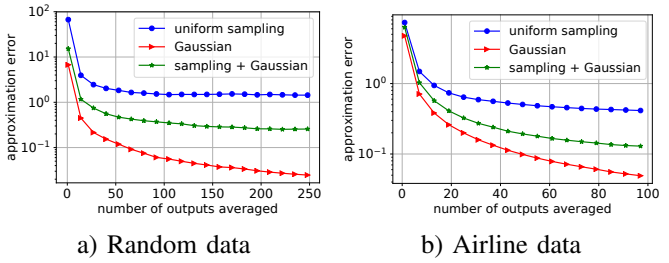


Fig. 8. Averaging for least norm problems. *Plot (a):* The parameters are $n = 50$, $d = 1000$, $m = 200$, $m' = 500$. *Plot (b):* Least norm averaging applied to a subset of the airline dataset. The parameters are $n = 2000$, $d = 11588$, $m = 4000$, $m' = 8000$. For this plot, the features include the pairwise interactions in addition to the original features.

parameters used in the experiments are $m = 10^4, m' = 10^5$ for plot a, and $m = 8 \times 10^3, m' = 8 \times 10^4$ for plot b. We observe that both plots in Figure 7 reveal similar trends where the hybrid approach leads to a lower approximation error but takes longer due to the additional processing required for SJLT.

E. Numerical Results for Least Norm Problems

Figure 8 shows the approximation error as a function of the number of averaged outputs in solving the least norm problem for two different datasets. This section numerically verifies the theoretical result for least norm problems using the right sketch method given in Section II-D. Note that the theoretical properties of this method are given in Theorem 2.13. The dataset for Figure 8(a) is randomly generated with dimensions $A \in \mathbb{R}^{50\times 1000}$. We observe that Gaussian sketch outperforms uniform sampling in terms of the approximation error. Furthermore, Figure 8(a) verifies that if we apply the hybrid approach of sampling first and then applying Gaussian sketch, then its performance falls between the extreme ends of only sampling and only using Gaussian sketch. Moreover, Figure 8(b) shows the results for the same experiment on a subset of the airline dataset where we have included the pairwise interactions as features which makes this an underdetermined linear system. Originally, we had 774 features for this dataset and if we include all $x_i x_j$ terms as features, we would have a total of 299925 features, most of which are zero for all samples. We have excluded the all-zero columns from this extended matrix to obtain the final dimensions 2000×11588.

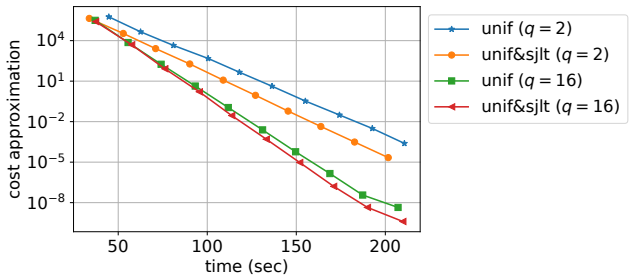


Fig. 9. Cost approximation $(f(x_t)-f(x^*))/f(x^*)$ vs time for the distributed IHS algorithm running on AWS Lambda to fit the linear least squares model on randomly generated data. Unif is short for uniform sampling and unif&sjlt is short for hybrid sketch where uniform sampling is followed by SJLT. Problem parameters are as follows: $n = 250000$, $d = 500$, $m = 6000$, $m' = 20000$, and q as specified in the legend.

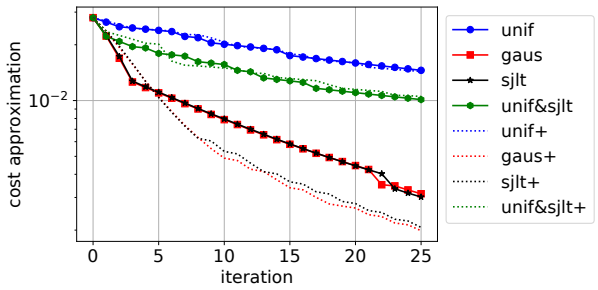


Fig. 10. Plot shows cost approximation of the iterate x_t (i.e., $(f(x_t) - f(x^*))/f(x^*)$) against iteration number t for various sketches in solving the log-barrier version of an inequality constrained optimization problem given in (37). Abbreviations used in the plot are as follows. Unif: Uniform sampling, gaus: Gaussian sketch, unif&sjlt: Hybrid sketch where uniform sampling is followed by SJLT. The abbreviations followed by $+$'s in the legend refer to the bias corrected versions. Problem parameters are as follows: $n = 500$, $d = 200$, $\lambda_1 = 1000$, $m = 50$, $m' = 8m = 400$, $q = 10$, $\lambda = 0.01$, $s = 10$.

F. Distributed Iterative Hessian Sketch

We have evaluated the performance of the distributed IHS algorithm on AWS Lambda. In the implementation, each serverless function is responsible for solving one sketched problem per iteration. Worker nodes wait idly once they finish their computation for that iteration until the next iterate x_{t+1} becomes available. The master node is implemented as another AWS Lambda function and is responsible for collecting and averaging the worker outputs and broadcasting the next iterate x_{t+1}.

Figure 9 shows the scaled difference between the cost for the t'th iterate and the optimal cost (i.e. $(f(x_t) - f(x^*))/f(x^*)$) against wall-clock time for the distributed IHS algorithm given in Algorithm 2. Due to the relatively small size of the problem, we have each worker compute the exact gradient without requiring an additional communication round per iteration to form the full gradient. We note that, for problems where it is not feasible for workers to form the full gradient due to limited memory, one can include an additional communication round where each worker sends their local gradient to the master node, and the master node forms the full gradient and distributes it to the worker nodes.

G. Inequality Constrained Optimization

Figure 10 compares the error performance of sketches with and without bias correction for the distributed Newton sketch algorithm when it is used to solve the log-barrier penalized problem given in (37). For each sketch, we have plotted the performance for $\lambda_2 = \lambda_1$ and the bias corrected version $\lambda_2 = \lambda_2^*$ (see Theorem 4.5). The bias corrected versions are shown as the dotted lines. In these experiments, we have set σ to the minimum of the singular values of DA as we have observed that setting σ to the minimum of the singular values of DA performed better than setting it to their mean.

Even though the bias correction formula is derived for Gaussian sketch, we observe that it improves the performance of SJLT as well. We see that Gaussian sketch and SJLT perform the best out of the 4 sketches we have experimented with. We note that computational complexity of sketching for SJLT is lower than it is for Gaussian sketch, and hence the natural choice would be to use SJLT in this case.

VIII. Discussion

In this work, we study averaging sketched solutions for linear least squares problems and averaging for randomized second order optimization algorithms. We discuss distributed sketching methods from the perspectives of convergence, bias, privacy, and distributed computing. Our results and numerical experiments suggest that distributed sketching methods offer a competitive straggler-resilient solution for solving large scale problems.

We have shown that for problems involving regularization, averaging requires a more detailed analysis compared to problems without regularization. When the regularization term is not scaled properly, the resulting estimators are biased, and averaging a large number of independent sketched solutions does not converge to the true solution. We have provided closed-form formulas for scaling the regularization coefficient to obtain unbiased estimators that guarantee convergence to the optimum.

Appendix A
Bounds for Other Sketching Matrices

In this section, we consider randomized Hadamard sketch, uniform sampling, and leverage score sampling for solving the unregularized linear least squares problem $\min_x f(x) = \|Ax - b\|_2^2$. For each of these sketching methods, we present upper bounds on the bias of the corresponding single sketch estimator. Then, we combine these in Theorem A.6, which provides high probability upper bounds for the error of the averaged estimator.

Lemma A.1 expresses the expected objective value difference in terms of the bias and variance of the single sketch estimator.

Lemma A.1: For any i.i.d. sketching matrices S_k, $k = 1, \dots, q$, the expected objective value difference for the averaged estimator $\bar{x}$ can be decomposed as

$$\mathbb{E}[f(\bar{x})] - f(x^*) = \\ = \frac{1}{q}\,\mathbb{E}\left[\|A\hat{x} - Ax^*\|_2^2\right] + \frac{q-1}{q}\|\,\mathbb{E}[A\hat{x}] - Ax^*\|_2^2, \quad (40)$$

where $\hat{x}$ is the single sketch estimator.

Lemma A.2 establishes an upper bound on the norm of the bias for any i.i.d. sketching matrix. The results in the remainder of this section will build on this lemma.

Lemma A.2: Let the SVD of A be denoted as $A = U\Sigma V^T$. Let $z := U^T S^T S b^\perp$ and $Q := (U^T S^T S U)^{-1}$ where $b^\perp := b - Ax^*$. Define the event E as $(1-\epsilon)I_d \preceq Q \preceq (1+\epsilon)I_d$. Then, for any sketch matrix S with $\mathbb{E}[S^T S] = I_n$, the bias of the single sketch estimator, conditioned on E, is upper bounded as

$$\| \mathbb{E}[A\hat{x}|E] - Ax^* \|_2 \leq \sqrt{4\epsilon \, \mathbb{E}[\|z\|_2^2 | E]}\,. \tag{41}$$

The event E is a high probability event and it is equivalent to the subspace embedding property (e.g. [2]). We will analyze the unconditioned error later in Theorem A.6.

We note that Lemma A.1 and A.2 apply to all of the sketching matrices considered in this work. We now give specific bounds for the bias of the single sketch estimator separately for each of randomized Hadamard sketch, uniform sampling, and leverage score sampling.

Randomized Hadamard sketch: This method has the advantage of low computational complexity due to the Fast Hadamard Transform, which can be computed in $O(n \log n)$, whereas applying the Gaussian sketch takes $O(mn)$ time for length n vectors. Lemma A.3 states the upper bound for the bias for randomized Hadamard sketch.

Lemma A.3: For randomized Hadamard sketch, the bias is upper bounded as

$$\| \mathbb{E}[A\hat{x}|E] - Ax^* \|_2 \leq \sqrt{4\epsilon \frac{d}{m} f(x^*)}. \tag{42}$$

Uniform sampling: We note that the bias of the uniform sampling estimator is different when it is computed with or without replacement. The reason for this is that the rows of the sketching matrix S for uniform sampling are independent in the case of sampling with replacement, which breaks down in the case of sampling without replacement. Lemma A.4 provides bounds for both cases.

Lemma A.4: For uniform sampling, the bias can be upper bounded as

$$\| \mathbb{E}[A\hat{x}|E] - Ax^* \|_2 \leq \sqrt{4\epsilon \frac{\mu d}{m} f(x^*)} \tag{43}$$

$$\| \mathbb{E}[A\hat{x}|E] - Ax^* \|_2 \leq \sqrt{4\epsilon \frac{\mu d}{m} \frac{n-m}{n-1} f(x^*)}, \tag{44}$$

for sampling with and without replacement, respectively, where $\tilde{u}_i \in \mathbb{R}^d$ denotes the i'th row of U.

Leverage score sampling: Recall that the row leverage scores of a matrix $A = U\Sigma V^T$ are computed via $\ell_i = \|\tilde{u}_i\|_2^2$ for $i = 1, \ldots, n$ where $\tilde{u}_i \in \mathbb{R}^d$ denotes the i'th row of U. Since leverage score sampling looks at the data to help guide its sampling strategy, it achieves a lower error compared to uniform sampling which treats all the samples equally. Lemma A.5 gives the upper bound for the bias of the leverage score sampling estimator.

Lemma A.5: For leverage score sampling, the bias can be upper bounded as

$$\| \mathbb{E}[A\hat{x}|E] - Ax^* \|_2 \leq \sqrt{4\epsilon \frac{d}{m} f(x^*)}. \tag{45}$$

The results given in Lemma A.3, A.4, and A.5 can be combined with Lemma A.1. In particular, the error of the averaged estimator contains the squared bias term which is scaled with $(q-1)/q$. For a distributed computing system with a large number of worker nodes q, the contribution of the bias term is very close to 1 while the variance term will vanish as it is scaled with $1/q$.

We now leverage our analysis of the bias upper bounds to establish upper bounds on the error of the averaged estimator. Theorem A.6 gives high probability error bounds for different sketch types when the sketch size is set accordingly.

Theorem A.6: Suppose that the sketch size is selected as

$$m \gtrsim \frac{d + \log(n)}{\epsilon^2} \log(qd/\delta) \text{ for randomized Hadamard sketch,}$$

$$m \gtrsim \frac{\mu d}{\epsilon^2} \log(qd/\delta) \text{ for uniform sampling,}$$

$$m \gtrsim \frac{d}{\epsilon^2} \log(qd/\delta) \text{ for leverage score sampling,}$$

for some $\epsilon \in (0, \frac{1}{4}]$ and $\delta > 0$. Here μ is the row coherence defined in Section I-C, that is, $\mu = n/d \max_i \|\tilde{u}_i\|_2^2$. Then, the relative optimality gap of the averaged estimator obeys the following upper bounds:

$$P\Big(\frac{f(\bar{x})}{f(x^*)} \leq 1 + \gamma\Big) \geq$$

$$1 - \delta - \frac{d}{\gamma m}\Big(\frac{(1+\epsilon)^2}{q} + 4\epsilon\Big)$$
for randomized Hadamard sketch,

$$1 - \delta - \frac{\mu d}{\gamma m}\Big(\frac{(1+\epsilon)^2}{q} + 4\epsilon\Big)$$
for uniform sampling with replacement,

$$1 - \delta - \frac{\mu d}{\gamma m}\frac{n-m}{n-1}\Big(\frac{(1+\epsilon)^2}{q} + 4\epsilon\Big)$$
for uniform sampling without replacement,

$$1 - \delta - \frac{d}{\gamma m}\Big(\frac{(1+\epsilon)^2}{q} + 4\epsilon\Big)$$
for leverage score sampling,

for any $\gamma > 0$.

Remarkably, the optimality ratio of the distributed sketching estimator converges to 1 as q or m gets large. This is different from the results of [2], which do not imply convergence to 1 as $q \to \infty$ due to an additional bias term. We note that the relative error of the uniform sampling sketch has a dependence on the row coherence unlike the randomized Hadamard and leverage score sampling sketch.

The main idea for the proof of Theorem A.6 involves using the decomposition given in Lemma A.1 along with the upper bounds for $\mathbb{E}[\|U^T S^T S b^\perp\|_2^2 | E]$. Then, we use Markov's inequality to find a high probability bound for the approximation quality of the averaged estimator, conditioned on the events

E_k, $k = 1, \ldots, q$ defined as $(1-\epsilon)I_d \preceq (U^T S_k^T S_k U)^{-1} \preceq (1+\epsilon)I_d$. This inequality is a direct result of the subspace embedding property and holds with high probability. The last component of the proof deals with removing the conditioning on the events E_k to find the upper bound for the relative error $f(\bar{x})/f(x^*)$.

APPENDIX B
PROOFS

This section contains the proofs for the lemmas and theorems stated in the paper.

A. Proofs of Theorems and Lemmas in Section II

Proof of Theorem 2.2: Since the Gaussian sketch estimator is unbiased (i.e., $\mathbb{E}[\hat{x}_k] = x^*$), Lemma A.1 reduces to $\mathbb{E}[f(\bar{x})] - f(x^*) = \frac{1}{q}\mathbb{E}[\|A\hat{x}_1 - Ax^*\|_2^2]$. By Lemma 2.1, the error of the averaged solution conditioned on the events that $E_k = A^T S_k^T S_k A \succ 0$, $\forall k = 1, \ldots, q$ can exactly be written as

$$\mathbb{E}[\|A(\bar{x} - x^*)\|_2^2 | E_1 \cap \cdots \cap E_q] = \frac{1}{q}\frac{d}{m-d-1} f(x^*).$$

Using Markov's inequality, it follows that

$$P(\|A(\bar{x} - x^*)\|_2^2 \geq a | E_1 \cap \cdots \cap E_q) \leq \frac{1}{qa}\frac{d}{m-d-1} f(x^*).$$

The LHS can be lower bounded as

$$\begin{aligned}
&\frac{P(\|A(\bar{x} - x^*)\|_2^2 \geq a \cap (\bigcap_{k=1}^q E_k))}{P(\bigcap_{k=1}^q E_k)} \\
&\geq \frac{P(\|A(\bar{x} - x^*)\|_2^2 \geq a) + P(\bigcap_{k=1}^q E_k) - 1}{P(\bigcap_{k=1}^q E_k)} \qquad (46) \\
&= \frac{P(\|A(\bar{x} - x^*)\|_2^2 \geq a) + P(E_1)^q - 1}{P(E_1)^q}, \qquad (47)
\end{aligned}$$

where we have used the identity $P(A \cap B) \geq P(A) + P(B) - 1$ in (46) and the independence of the events E_k in (47). It follows

$$P(\|A(\bar{x} - x^*)\|_2^2 \leq a) \geq P(E_1)^q \left(1 - \frac{1}{qa}\frac{d}{m-d-1} f(x^*)\right).$$

Setting $a = f(x^*)\frac{\epsilon}{q}$ and plugging in $P(E_1) = 1$, which holds for $m \geq d$, we obtain

$$P\left(\frac{\|A(\bar{x} - x^*)\|_2^2}{f(x^*)} \leq \frac{\epsilon}{q}\right) \geq \left(1 - \frac{d/\epsilon}{m-d-1}\right).$$

■

Lemma B.1: For the Gaussian sketching matrix S and the matrix U whose columns are orthonormal, the smallest singular value of the product SU is bounded as:

$$P(\sigma_{\min}(SU) \leq 1 - \sqrt{d/m} - \delta) \leq \exp(-m\delta^2/2). \qquad (48)$$

This result follows from the concentration of Lipshitz functions of Gaussian random variables.

Proof of Lemma 2.7: We start by invoking a result on the concentration of Stieltjes transforms, namely Lemma 6 of [33]. This implies that

$$\begin{aligned}
&P\big(|\operatorname{tr}(U^T S^T SU - \epsilon i I)^{-1} - \mathbb{E}[\operatorname{tr}(U^T S^T SU - \epsilon i I)^{-1}]| > t\big) \\
&\quad \leq 4\exp\big(-t^2\epsilon^2/(16m)\big) \qquad (49)
\end{aligned}$$

where $i = \sqrt{-1}$. The matrix $U^T S^T SU$ can be written as a sum of rank-1 matrices as follows:

$$U^T S^T SU = \sum_{i=1}^m (\tilde{s}_i^T U)^T (\tilde{s}_i^T U) \qquad (50)$$

where $\tilde{s}_i^T \in \mathbb{R}^{1\times n}$ is the i'th row of S. The vectors $(\tilde{s}_i^T U)^T$ are independent as required by Lemma 6 of [33]. Next, we will bound the difference between the trace terms $\operatorname{tr}((U^T S^T SU)^{-1})$ and $\operatorname{tr}((U^T S^T SU - \epsilon i I)^{-1})$. First, we let the SVD decomposition of SU be $\tilde{U}\tilde{\Sigma}\tilde{V}^T$. Then, we have the following relations:

$$\begin{aligned}
SU &= \tilde{U}\tilde{\Sigma}\tilde{V}^T \\
U^T S^T SU &= \tilde{V}\tilde{\Sigma}^2\tilde{V}^T \\
(U^T S^T SU)^{-1} &= \tilde{V}\tilde{\Sigma}^{-2}\tilde{V}^T \\
U^T S^T SU - \epsilon i I &= \tilde{V}\operatorname{diag}(\tilde{\sigma}_j^2 - \epsilon i)\tilde{V}^T \\
(U^T S^T SU - \epsilon i I)^{-1} &= \tilde{V}\operatorname{diag}\left(\frac{1}{\tilde{\sigma}_j^2 - \epsilon i}\right)\tilde{V}^T \qquad (51)
\end{aligned}$$

where the notation $\operatorname{diag}(\tilde{\sigma}_j^2)$ refers to a diagonal matrix with diagonal entries equal to $\tilde{\sigma}_1^2, \tilde{\sigma}_2^2, \ldots$. The difference between the trace terms can be bounded as follows:

$$\begin{aligned}
&\big|\operatorname{tr}((U^T S^T SU)^{-1}) - \operatorname{tr}((U^T S^T SU - \epsilon i I)^{-1})\big| = \\
&= \left|\operatorname{tr}\left(\tilde{V}\operatorname{diag}\left(\frac{1}{\tilde{\sigma}_j^2} - \frac{1}{\tilde{\sigma}_j^2 - \epsilon i}\right)\tilde{V}^T\right)\right| \\
&= \left|\sum_{j=1}^d \left(\frac{1}{\tilde{\sigma}_j^2} - \frac{1}{\tilde{\sigma}_j^2 - \epsilon i}\right)\right| = \left|\sum_{j=1}^d \left(\frac{-\epsilon i}{\tilde{\sigma}_j^2(\tilde{\sigma}_j^2 - \epsilon i)}\right)\right| \\
&\leq \sum_{j=1}^d \left|\frac{-\epsilon i}{\tilde{\sigma}_j^2(\tilde{\sigma}_j^2 - \epsilon i)}\right| = \sum_{j=1}^d \frac{\epsilon}{\tilde{\sigma}_j^2|\tilde{\sigma}_j^2 - \epsilon i|} \\
&\leq \sum_{j=1}^d \frac{\epsilon}{\tilde{\sigma}_j^4} \leq \sum_{j=1}^d \frac{\epsilon}{\tilde{\sigma}_j^2\sigma_{\min}^2(SU)} \\
&= \frac{\epsilon}{\sigma_{\min}^2(SU)}\sum_{j=1}^d \frac{1}{\tilde{\sigma}_j^2} = \frac{\epsilon}{\sigma_{\min}^2(SU)}\operatorname{tr}\left((U^T S^T SU)^{-1}\right). \qquad (52)
\end{aligned}$$

Equivalently, we have

$$\begin{aligned}
&\big(1 - \frac{\epsilon}{\sigma_{\min}^2(SU)}\big)\operatorname{tr}((U^T S^T SU)^{-1}) \\
&\leq \operatorname{tr}((U^T S^T SU - \epsilon i I)^{-1}) \\
&\leq \big(1 + \frac{\epsilon}{\sigma_{\min}^2(SU)}\big)\operatorname{tr}((U^T S^T SU)^{-1}). \qquad (53)
\end{aligned}$$

The relations in (49) and (53) together imply the following concentration result:

$$\begin{aligned}
&P\big(|\operatorname{tr}((U^T S^T SU)^{-1}) - \mathbb{E}[\operatorname{tr}((U^T S^T SU)^{-1})]| \leq t \\
&+ \frac{\epsilon}{\sigma_{\min}^2(SU)}\left(\operatorname{tr}((U^T S^T SU)^{-1}) + \mathbb{E}[\operatorname{tr}((U^T S^T SU)^{-1})]\right)\big) \\
&\geq 1 - 4\exp\big(-t^2\epsilon^2/(16m)\big). \qquad (54)
\end{aligned}$$

We know that $\mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})] = \frac{md}{m-d-1}$. Using the upper bound $\mathrm{tr}((U^TS^TSU)^{-1}) \leq d\sigma_{\min}^{-2}(SU)$ yields

$$P\Big(|\,\mathrm{tr}((U^TS^TSU)^{-1}) - \mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})]| \leq t + \frac{\epsilon}{\sigma_{\min}^2(SU)}\big(\frac{d}{\sigma_{\min}^2(SU)} + \frac{md}{m-d-1}\big)\Big) \geq 1 - 4\exp\left(-t^2\epsilon^2/(16m)\right). \quad (55)$$

The concentration inequality in (55) and the concentration of the minimum singular value $P(\frac{1}{\sigma_{\min}(SU)} \leq \frac{1}{1-\sqrt{d/m}-\delta}) \geq 1 - \exp(-m\delta^2/2)$ from Lemma B.1 together imply that

$$P\Big(|\,\mathrm{tr}((U^TS^TSU)^{-1}) - \mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})]| \leq t + \frac{\epsilon}{(1-\sqrt{d/m}-\delta)^2}\big(\frac{d}{(1-\sqrt{d/m}-\delta)^2} + \frac{md}{m-d-1}\big)\Big) \geq 1 - 4\exp\left(-t^2\epsilon^2/(16m)\right) - \exp(-m\delta^2/2). \quad (56)$$

We now simplify this bound by observing that

$$\begin{aligned}
&\frac{d}{(1-\sqrt{d/m}-\delta)^2} = \\
&= \frac{d}{(1+\frac{d}{m}-2\sqrt{\frac{d}{m}}) + \delta^2 - 2\delta(1-\sqrt{\frac{d}{m}})} \\
&= \frac{md}{m+d-2\sqrt{md}+\delta^2 m - 2\delta m(1-\sqrt{\frac{d}{m}})} \\
&\geq \frac{md}{m-d(2\sqrt{\alpha}-1)} \\
&\geq \frac{md}{m-d-1}
\end{aligned} \quad (57)$$

where we used $m \geq \alpha d$ for the first inequality. We have also used the following simple inequality

$$2\delta m(1-\sqrt{d/m}) \geq 2\delta m(1-\frac{1}{\sqrt{2}}) \geq \delta m \geq \delta^2 m\,. \quad (58)$$

For the fourth line to be true, we need $m - 2d\sqrt{\alpha} + d \leq m-d-1$. This is satisfied if $1 \leq 2d(\sqrt{\alpha}-1)$ or $\alpha \geq (\frac{1}{2d}+1)^2$. We note that this is already implied by our assumption that $m \gtrsim d$. Using the above observation, we simplify the bound as follows:

$$P\Big(|\,\mathrm{tr}((U^TS^TSU)^{-1}) - \mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})]| \leq t + \frac{2d\epsilon}{(1-\sqrt{d/m}-\delta)^4}\Big) \geq 1 - 4\exp\left(-t^2\epsilon^2/(16m)\right) - \exp(-m\delta^2/2). \quad (59)$$

Scaling $\epsilon \leftarrow \epsilon\frac{(1-\sqrt{d/m}-\delta)^4}{2d}$ gives

$$P\Big(|\,\mathrm{tr}((U^TS^TSU)^{-1}) - \mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})]| \leq t + \epsilon\Big) \geq 1 - 4e^{-\frac{t^2\epsilon^2(1-\sqrt{d/m}-\delta)^8}{64md^2}} - e^{-m\delta^2/2}. \quad (60)$$

Setting $t = \epsilon$ and then $\epsilon \leftarrow \epsilon/2$ give us the final result:

$$P\Big(|\,\mathrm{tr}((U^TS^TSU)^{-1}) - \mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})]| \leq \epsilon\Big) \geq 1 - 4e^{-\frac{\epsilon^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} - e^{-m\delta^2/2}. \quad (61)$$

■

Proof of Theorem 2.4: Let us consider the single sketch estimator $\hat{x}$. The relative error of $\hat{x}$ can be expressed as

$$\begin{aligned}
f(\hat{x}) - f(x^*) &= \|A(\hat{x}-x^*)\|_2^2 \\
&= \|A(A^TS^TSA)^{-1}A^TS^TSb^\perp\|_2^2 \\
&= \|A(SA)^\dagger Sb^\perp\|_2^2
\end{aligned} \quad (62)$$

where the superscript † indicates the pseudo-inverse. We will rewrite the error expression as:

$$f(\hat{x}) - f(x^*) = \left\|\frac{\|b^\perp\|_2}{\sqrt{m}}A(SA)^\dagger\frac{\sqrt{m}}{\|b^\perp\|_2}Sb^\perp\right\|_2^2 \quad (63)$$

and define $z := \frac{\sqrt{m}}{\|b^\perp\|_2}Sb^\perp$ and $M := \frac{\|b^\perp\|_2}{\sqrt{m}}A(SA)^\dagger$. This simplifies the previous expression

$$f(\hat{x}) - f(x^*) = \|Mz\|_2^2 = z^TM^TMz. \quad (64)$$

Lemma 2.5 implies

$$P\Big(f(\hat{x}) - f(x^*) - \mathbb{E}_{Sb^\perp}[f(\hat{x}) - f(x^*)] > 2\|M^TM\|_F\sqrt{\epsilon} + 2\|M^TM\|_2\epsilon\Big|SU\Big) \leq e^{-\epsilon}. \quad (65)$$

The terms $\|M^TM\|_F$ and $\|M^TM\|_2$ reduce to the following expressions:

$$\|M^TM\|_F = \frac{f(x^*)}{m}\sqrt{\mathrm{tr}((U^TS^TSU)^{-2})} \leq \frac{f(x^*)}{m}\frac{\sqrt{d}}{\sigma_{\min}^2(SU)} \quad (66)$$

$$\|M^TM\|_2 = \frac{f(x^*)}{m}\frac{1}{\sigma_{\min}^2(SU)}\,. \quad (67)$$

Plugging these in (65) and taking the expectation of both sides with respect to SU give us

$$P\Big(f(\hat{x}) - f(x^*) > \frac{f(x^*)}{m}\,\mathrm{tr}((U^TS^TSU)^{-1}) + 2\frac{f(x^*)}{m\sigma_{\min}^2(SU)}(\sqrt{\epsilon d}+\epsilon)\Big) \leq e^{-\epsilon}. \quad (68)$$

Next, we combine (68) with Lemma 2.7 and the concentration of the minimum singular value from Lemma B.1 to obtain

$$\begin{aligned}
&P\Big(f(\hat{x}) - f(x^*) > \frac{f(x^*)}{m}\,\mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})] \\
&\qquad + 2\frac{f(x^*)}{m(1-\sqrt{d/m}-s)^2}(\sqrt{\epsilon d}+\epsilon) + \frac{f(x^*)}{m}\gamma\Big) \\
&\leq e^{-\epsilon} + 4e^{-\frac{\gamma^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} + e^{-m\delta^2/2} + e^{-ms^2/2}.
\end{aligned} \quad (69)$$

Assuming that $\sqrt{\epsilon d} > \epsilon$, we can write:

$$\begin{aligned}
&P\Big(f(\hat{x}) - f(x^*) > \frac{f(x^*)}{m}\,\mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})] \\
&\qquad + \frac{4f(x^*)\sqrt{\epsilon d}}{m(1-\sqrt{d/m}-s)^2} + \frac{f(x^*)}{m}\gamma\Big) \\
&\leq e^{-\epsilon} + 4e^{-\frac{\gamma^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} + e^{-m\delta^2/2} + e^{-ms^2/2}.
\end{aligned} \quad (70)$$

Making the change of variable $\epsilon \leftarrow \epsilon^2/d$ yields:

$$P\Big(f(\hat{x}) - f(x^*) > \frac{f(x^*)}{m}\,\mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})] + \frac{4f(x^*)\epsilon}{m(1-\sqrt{d/m}-s)^2} + \frac{f(x^*)}{m}\gamma\Big) \leq e^{-\epsilon^2/d} + 4e^{-\frac{\gamma^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} + e^{-m\delta^2/2} + e^{-ms^2/2}. \quad (71)$$

Scaling $\epsilon \leftarrow \epsilon\frac{(1-\sqrt{d/m}-s)^2}{4}$ yields:

$$P\Big(f(\hat{x}) - f(x^*) > \frac{f(x^*)}{m}\,\mathbb{E}[\mathrm{tr}((U^TS^TSU)^{-1})] + \frac{f(x^*)(\epsilon+\gamma)}{m}\Big) \leq e^{-\frac{\epsilon^2(1-\sqrt{d/m}-s)^4}{16d}} + 4e^{-\frac{\gamma^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} + e^{-m\delta^2/2} + e^{-ms^2/2}\,. \quad (72)$$

Letting $\gamma = \epsilon$ and $s = \delta$, and then scaling $\epsilon \leftarrow \epsilon m/2$ lead to

$$P\big(\frac{f(\hat{x})}{f(x^*)} > \frac{m-1}{m-d-1} + \epsilon\big) \leq e^{-\frac{\epsilon^2m^2(1-\sqrt{d/m}-\delta)^4}{64d}} + 4e^{-\frac{\epsilon^4m^4(1-\sqrt{d/m}-\delta)^8}{2^{14}md^2}} + 2e^{-m\delta^2/2}. \quad (73)$$

Let us also set $\delta = \epsilon$:

$$P\big(\frac{f(\hat{x})}{f(x^*)} > \frac{m-1}{m-d-1} + \epsilon\big) \leq e^{-\frac{\epsilon^2m^2(1-\sqrt{d/m}-\epsilon)^4}{64d}} + 4e^{-\frac{\epsilon^4m^4(1-\sqrt{d/m}-\epsilon)^8}{2^{14}md^2}} + 2e^{-m\epsilon^2/2}. \quad (74)$$

Under our assumption that $m \gtrsim d$, the above probability is bounded by

$$P\big(\frac{f(\hat{x})}{f(x^*)} > \frac{m-1}{m-d-1} + \epsilon\big) \leq C_1e^{-C_2\epsilon^4m}. \quad (75)$$

We now move on to find a lower bound. We note that Lemma 2.5 can be used as follows to find a lower bound

$$P\big(z^T(-G)z - \mathbb{E}[z^T(-G)z] > 2\|(-G)\|_F\sqrt{\epsilon} + 2\|(-G)\|_2\epsilon\big) = P\big(z^TGz < \mathbb{E}[z^TGz] - 2\|G\|_F\sqrt{\epsilon} - 2\|G\|_2\epsilon\big) \leq e^{-\epsilon}. \quad (76)$$

Therefore, we have the following lower bound for the error:

$$P\Big(f(\hat{x}) - f(x^*) < \mathbb{E}_{Sb^\perp}[f(\hat{x}) - f(x^*)] - 2\|M^TM\|_F\sqrt{\epsilon} - 2\|M^TM\|_2\epsilon\Big|SU\Big) \leq e^{-\epsilon}. \quad (77)$$

Using the upper bound for $\|M^TM\|_F$ and the exact expression for $\|M^TM\|_2$, and taking the expectation with respect to SU, we obtain the following

$$P\Big(f(\hat{x}) - f(x^*) < \frac{f(x^*)}{m}\,\mathrm{tr}((U^TS^TSU)^{-1}) - 2\frac{f(x^*)}{m\sigma^2_{\min}(SU)}(\sqrt{\epsilon d} + \epsilon)\Big) \leq e^{-\epsilon}. \quad (78)$$

Applying the same steps as before, we arrive at the following bound:

$$P\big(\frac{f(\hat{x})}{f(x^*)} < \frac{m-1}{m-d-1} - \epsilon\big) \leq C_1e^{-C_2\epsilon^4m}. \quad (79)$$

■

Proof of Theorem 2.8: Consider the averaged estimator $\bar{x} = \frac{1}{q}\sum_{k=1}^q \hat{x}_k$:

$$f(\bar{x}) - f(x^*) = \|A(\bar{x}-x^*)\|_2^2 = \Big\|\frac{1}{q}\sum_{k=1}^q A(S_kA)^\dagger S_kb^\perp\Big\|_2^2 = \Big\|\frac{1}{q}\sum_{k=1}^q M_kz_k\Big\|_2^2 \quad (80)$$

where we defined

$$M_k := \frac{\|b^\perp\|_2}{\sqrt{m}}A(S_kA)^\dagger, \quad z_k := \frac{\sqrt{m}}{\|b^\perp\|_2}S_kb^\perp. \quad (81)$$

We note that the entries of the vector z are distributed as i.i.d. standard normal. We can manipulate the error expression as follows

$$\sum_{k=1}^q M_kz_k = \begin{bmatrix}M_1 & \dots & M_q\end{bmatrix}\begin{bmatrix}z_1\\ \vdots \\ z_q\end{bmatrix} = Mz \quad (82)$$

and then

$$f(\bar{x}) - f(x^*) = \frac{1}{q^2}\|Mz\|_2^2 = z^T\big(\frac{1}{q^2}M^TM\big)z. \quad (83)$$

We will now find an equivalent expression for $\|\frac{1}{q^2}M^TM\|_F^2 = \frac{1}{q^4}\|M^TM\|_F^2$:

$$\begin{aligned}
&\|M^TM\|_F^2 = \\
&= \mathrm{tr}(M^TMM^TM) = \mathrm{tr}(MM^TMM^T) \\
&= \mathrm{tr}\left(\sum_{k=1}^q M_kM_k^T\sum_{l=1}^q M_lM_l^T\right) \\
&= \frac{f^2(x^*)}{m^2}\sum_{k=1}^q\sum_{l=1}^q \mathrm{tr}\left((S_kU)^{\dagger T}(S_kU)^\dagger(S_lU)^{\dagger T}(S_lU)^\dagger\right) \\
&= \frac{f^2(x^*)}{m^2}\Big(\sum_{k=1}^q \mathrm{tr}((U^TS_k^TS_kU)^{-2}) \\
&\quad + \sum_{k\neq l}\mathrm{tr}\left((S_kU)^{\dagger T}(S_kU)^\dagger(S_lU)^{\dagger T}(S_lU)^\dagger\right)\Big) \\
&\leq \frac{f^2(x^*)}{m^2}\Big(\sum_{k=1}^q \frac{d}{\sigma^4_{\min}(S_kU)} \\
&\quad + \sum_{k\neq l}\sqrt{\mathrm{tr}\left((S_kU)^{\dagger T}(S_kU)^\dagger\right)}\sqrt{\mathrm{tr}\left((S_lU)^{\dagger T}(S_lU)^\dagger\right)}\Big) \\
&= \frac{f^2(x^*)}{m^2}\Big(\sum_{k=1}^q \frac{d}{\sigma^4_{\min}(S_kU)} \\
&\quad + \sum_{k\neq l}\sqrt{\mathrm{tr}\left((U^TS_k^TS_kU)^{-1}\right)}\sqrt{\mathrm{tr}\left((U^TS_l^TS_lU)^{-1}\right)}\Big) \\
&\leq \frac{f^2(x^*)}{m^2}\Big(\sum_{k=1}^q \frac{d}{\sigma^4_{\min}(S_kU)} + \sum_{k\neq l}\frac{d}{\sigma_{\min}(S_kU)\sigma_{\min}(S_lU)}\Big)
\end{aligned} \quad (84)$$

where we have used the Cauchy-Schwarz inequality to bound the trace of the matrix product.

Next, we look at $\|\frac{1}{q^2}M^TM\|_2 = \frac{1}{q^2}\|M^TM\|_2$:

$$\begin{aligned}\|M^TM\|_2 = \|M\|_2^2 = \|MM^T\|_2 &= \left\|\sum_{k=1}^q M_kM_k^T\right\|_2 \\ &= \frac{\|b^\perp\|_2^2}{m}\left\|\sum_{k=1}^q (S_kU)^{\dagger^T}(S_kU)^\dagger\right\|_2 \\ &\le \frac{\|b^\perp\|_2^2}{m}\sum_{k=1}^q \|(S_kU)^{\dagger^T}(S_kU)^\dagger\|_2 \\ &= \frac{\|b^\perp\|_2^2}{m}\sum_{k=1}^q \frac{1}{\sigma_{\min}^2(S_kU)}. \end{aligned} \tag{85}$$

By Lemma 2.5, we have

$$\begin{aligned}P\Big(f(\bar{x}) - f(x^*) - \mathbb{E}_{S_kb^\perp}[f(\bar{x}) - f(x^*)] \\ > \frac{2}{q^2}\|M^TM\|_F\sqrt{\epsilon} + \frac{2}{q^2}\|M^TM\|_2\epsilon\Big|SU_k\Big) \le e^{-\epsilon}.\end{aligned} \tag{86}$$

We can take expectation of both sides with respect to SU_k, $k = 1, \ldots, q$ to remove the conditioning. The expectation term inside the probability is equal to

$$\begin{aligned}\mathbb{E}_{S_kb^\perp}[f(\bar{x}) - f(x^*)] = \operatorname{tr}(\frac{1}{q^2}M^TM) &= \frac{1}{q^2}\operatorname{tr}(MM^T) \\ &= \frac{f(x^*)}{mq^2}\sum_{k=1}^q \operatorname{tr}((U^TS_k^TS_kU)^{-1}).\end{aligned} \tag{87}$$

Hence, we have

$$\begin{aligned}P\Big(f(\bar{x}) - f(x^*) - \frac{f(x^*)}{mq^2}\sum_{k=1}^q \operatorname{tr}((U^TS_k^TS_kU)^{-1}) \\ > \frac{2}{q^2}\|M^TM\|_F\sqrt{\epsilon} + \frac{2}{q^2}\|M^TM\|_2\epsilon\Big) \le e^{-\epsilon}.\end{aligned} \tag{88}$$

We now use the concentration bound of the trace term in Lemma 2.7 to obtain

$$\begin{aligned}P\Big(f(\bar{x}) - f(x^*) - \frac{f(x^*)}{mq^2}\sum_{k=1}^q \mathbb{E}[\operatorname{tr}((U^TS_k^TS_kU)^{-1})] \\ > \frac{2}{q^2}\|M^TM\|_F\sqrt{\epsilon} + \frac{2}{q^2}\|M^TM\|_2\epsilon + \frac{f(x^*)}{mq^2}q\gamma\Big) \\ \le e^{-\epsilon} + 1 - \big(1 - 4e^{-\frac{\gamma^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} - e^{-m\delta^2/2}\big)^q\end{aligned} \tag{89}$$

where we have used the independence of S_k's. Using the upper bounds for $\|M^TM\|_F$ and $\|M^TM\|_2$, we make the following observation. If the minimum singular values of all of S_kU, $k = 1, \ldots, q$ satisfy $\sigma_{\min}(S_kU) > 1 - \sqrt{d/m} - s$, which occurs with probability at least $(1 - e^{-ms^2/2})^q$ from Lemma B.1, then the following holds

$$\begin{aligned}&\|M^TM\|_F\sqrt{\epsilon} + \|M^TM\|_2\epsilon \\ &\le \frac{f(x^*)}{m}\Bigg(\sqrt{\frac{qd}{(1-\sqrt{d/m}-s)^4} + \frac{(q^2-q)d}{(1-\sqrt{d/m}-s)^2}}\sqrt{\epsilon} \\ &\quad + \frac{q\epsilon}{(1-\sqrt{d/m}-s)^2}\Bigg) \\ &\le \frac{f(x^*)}{m}\frac{q(\sqrt{\epsilon d}+\epsilon)}{(1-\sqrt{d/m}-s)^2} \le \frac{f(x^*)}{m}\frac{2q\sqrt{\epsilon d}}{(1-\sqrt{d/m}-s)^2}\end{aligned} \tag{90}$$

where we used $1-\sqrt{d/m}-s < 1$ and $\epsilon \le \sqrt{\epsilon d}$ for the last two inequalities. Combining this observation with the probability bound, we arrive at

$$\begin{aligned}&P\Big(f(\bar{x}) - f(x^*) - \frac{f(x^*)}{mq^2}\sum_{k=1}^q \mathbb{E}[\operatorname{tr}((U^TS_k^TS_kU)^{-1})] \\ &\qquad > \frac{2}{q^2}\frac{f(x^*)}{m}\frac{2q\sqrt{\epsilon d}}{(1-\sqrt{d/m}-s)^2} + \frac{f(x^*)}{mq^2}q\gamma\Big) \\ &\le e^{-\epsilon} + 1 - \big(1 - 4e^{-\frac{\gamma^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} - e^{-m\delta^2/2}\big)^q \\ &\quad + 1 - (1 - e^{-ms^2/2})^q.\end{aligned} \tag{91}$$

Simplifying the expressions and making the change $\epsilon \leftarrow \epsilon^2/d$ lead to

$$\begin{aligned}&P\Big(f(\bar{x}) - f(x^*) - \frac{f(x^*)}{q}\frac{d}{m-d-1} \\ &\qquad > \frac{4f(x^*)}{mq}\frac{\epsilon}{(1-\sqrt{d/m}-s)^2} + \frac{f(x^*)}{mq}\gamma\Big) \\ &\le e^{-\epsilon^2/d} + 2 - \big(1 - 4e^{-\frac{\gamma^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} - e^{-m\delta^2/2}\big)^q \\ &\quad - (1 - e^{-ms^2/2})^q.\end{aligned} \tag{92}$$

Next, we let $\epsilon \leftarrow \epsilon\frac{(1-\sqrt{d/m}-s)^2}{4}$ and obtain

$$\begin{aligned}&P\Big(f(\bar{x}) - f(x^*) - \frac{f(x^*)}{q}\frac{d}{m-d-1} > \frac{f(x^*)}{mq}(\epsilon+\gamma)\Big) \\ &\le e^{-\frac{\epsilon^2(1-\sqrt{d/m}-s)^4}{16d}} + 2 \\ &\quad - (1 - e^{-ms^2/2})^q - \big(1 - 4e^{-\frac{\gamma^4(1-\sqrt{d/m}-\delta)^8}{2^{10}md^2}} - e^{-m\delta^2/2}\big)^q.\end{aligned} \tag{93}$$

We now set $\gamma = \epsilon$ and scale $\epsilon \leftarrow \epsilon\frac{mq}{2}$:

$$\begin{aligned}&P\Big(\frac{f(\bar{x})}{f(x^*)} > 1 + \frac{1}{q}\frac{d}{m-d-1} + \epsilon\Big) \le e^{-\frac{\epsilon^2m^2q^2(1-\sqrt{d/m}-s)^4}{64d}} \\ &+ 2 - (1 - e^{-ms^2/2})^q \\ &- \big(1 - 4e^{-\frac{\epsilon^4m^4q^4(1-\sqrt{d/m}-\delta)^8}{2^{14}md^2}} - e^{-m\delta^2/2}\big)^q.\end{aligned} \tag{94}$$

Let us use the assumption $m \gtrsim d$ to further simplify:

$$\begin{aligned}&P\Big(\frac{f(\bar{x})}{f(x^*)} > 1 + \frac{1}{q}\frac{d}{m-d-1} + \epsilon\Big) \\ &\qquad \le C_1e^{-C_2(q\epsilon)^2m} + 2 - (1 - e^{-ms^2/2})^q \\ &\qquad\quad - \big(1 - 4e^{-\frac{\epsilon^4m^4q^4(1-\sqrt{d/m}-\delta)^8}{2^{14}md^2}} - e^{-m\delta^2/2}\big)^q.\end{aligned} \tag{95}$$

We can use Bernoulli's inequality $(1-e^{-ms^2/2})^q \geq 1 - qe^{-ms^2/2}$ and arrive at:

$$P\Big(\frac{f(\bar{x})}{f(x^*)} > 1 + \frac{1}{q}\frac{d}{m-d-1} + \epsilon\Big) \leq C_1 e^{-C_2(q\epsilon)^2 m} + C_3 q e^{-C_4(q\epsilon)^4 m(1-\sqrt{d/m}-s)^8} + 2qe^{-ms^2/2} \tag{96}$$

where we also set $\delta = s$. Picking $s = q\epsilon$ yields:

$$P\Big(\frac{f(\bar{x})}{f(x^*)} > 1 + \frac{1}{q}\frac{d}{m-d-1} + \epsilon\Big) \leq C_1 e^{-C_2(q\epsilon)^2 m} + C_3 q e^{-C_4(q\epsilon)^4 m(1-\sqrt{d/m}-q\epsilon)^8} + 2qe^{-m(q\epsilon)^2/2}. \tag{97}$$

Finally, we obtain the following simpler expression for the bound:

$$P\Big(\frac{f(\bar{x})}{f(x^*)} > 1 + \frac{1}{q}\frac{d}{m-d-1} + \epsilon\Big) \leq qC_1 e^{-C_2(q\epsilon)^4 m} \tag{98}$$

where we redefine the constants C_1, C_2.

Lower bound can be obtained by applying the same steps:

$$P\Big(\frac{f(\bar{x})}{f(x^*)} < 1 + \frac{1}{q}\frac{d}{m-d-1} - \epsilon\Big) \leq qC_1 e^{-C_2(q\epsilon)^4 m}. \tag{99}$$

■

Proof of Theorem 2.11: i) For unbiased estimators: We begin by invoking Lemma A.1 which gives us

$$\mathbb{E}[\|A(\bar{x}-x^*)\|_2^2] = \frac{1}{q}\,\mathbb{E}[\|A(\hat{x}_1-x^*)\|_2^2] \tag{100}$$

since we assume that $\hat{x}_k$'s are unbiased, i.e., $\mathbb{E}[\hat{x}_k] = x^*$ for $k = 1, \ldots, q$.

This shows that the error of the averaged estimator is equal to the error of the single sketch estimator scaled by $\frac{1}{q}$. Hence, we can leverage the lower bound result for the single sketch estimator. We now give the details of how to find the lower bound for the single sketch estimator which was first shown in [31].

The Fisher information matrix for estimating x^* from Sb can be constructed as follows:

$$I(Sb;x^*) = \mathbb{E}_{Sb}[\nabla_{x^*}\log g(Sb;x^*)\nabla_{x^*}\log g(Sb;x^*)^T] \tag{101}$$

where $g(Sb;x^*)$ is the probability density function of Sb conditioned on SA, i.e., $g(Sb;x^*)$ is the multivariate Gaussian distribution with $\mathcal{N}(SAx^*, \frac{1}{m}\|b^\perp\|_2^2 I_m)$. We recall the definition that $b^\perp := b - Ax^*$.

The gradient of the logarithm of the probability density function with respect to x^* is

$$\nabla_{x^*}\log g(Sb;x^*) = \frac{m}{\|b^\perp\|_2^2}(SA)^T(Sb-SAx^*). \tag{102}$$

Plugging this in (101) yields

$$\begin{aligned} I(Sb;x^*) &= \\ &= \frac{m^2}{\|b^\perp\|_2^4}\,\mathbb{E}_{Sb}\left[(SA)^T(Sb-SAx^*)(Sb-SAx^*)^T SA\right] \\ &= \frac{m^2}{\|b^\perp\|_2^4}(SA)^T\big(\frac{\|b^\perp\|_2^2}{m}I_m\big)SA \\ &= \frac{m}{\|b^\perp\|_2^2}A^T S^T SA. \end{aligned} \tag{103}$$

We note that the expectation of the error conditioned on SA is equivalent to

$$\begin{aligned} \mathbb{E}[\|A(\hat{x}-x^*)\|_2^2|SA] &= \mathbb{E}[\mathrm{tr}(A(\hat{x}-x^*)(\hat{x}-x^*)^T A^T)|SA] \\ &= \mathrm{tr}\left(A\,\mathbb{E}[(\hat{x}-x^*)(\hat{x}-x^*)^T|SA]A^T\right) \\ &\geq \frac{\|b^\perp\|_2^2}{m}\,\mathrm{tr}(A(A^TS^TSA)^{-1}A^T), \end{aligned} \tag{104}$$

where the last line follows from the Cramér lower bound [42] given by

$$\mathbb{E}[(\hat{x}-x^*)(\hat{x}-x^*)^T|SA] \succeq I^{-1}(Sb;x^*). \tag{105}$$

Taking the expectation of both sides with respect to SA yields

$$\mathbb{E}[\|A(\hat{x}-x^*)\|_2^2] \geq f(x^*)\frac{d}{m-d-1}. \tag{106}$$

Hence, we arrive at

$$\mathbb{E}[\|A(\bar{x}-x^*)\|_2^2] \geq \frac{f(x^*)}{q}\frac{d}{m-d-1}. \tag{107}$$

ii) For any estimator: We again invoke Lemma A.1, which gives us

$$\begin{aligned} &\mathbb{E}[\|A(\bar{x}-x^*)\|_2^2] = \\ &= \frac{1}{q}\,\mathbb{E}[\|A(\hat{x}_1-x^*)\|_2^2] + \frac{q-1}{q}\|\,\mathbb{E}[A\hat{x}_1] - Ax^*\|_2^2 \\ &\geq \frac{f(x^*)}{q}\frac{d}{m} + \frac{q-1}{q}\|\,\mathbb{E}[A\hat{x}_1] - Ax^*\|_2^2 \\ &\geq \frac{f(x^*)}{q}\frac{d}{m} \end{aligned} \tag{108}$$

where the first inequality follows from Lemma 2.10. ■

B. Proofs of Theorems and Lemmas in Section III

Proof of Theorem 3.2: The update rule for distributed IHS is given as

$$x_{t+1} = x_t - \mu\frac{1}{q}\sum_{k=1}^{q}(A^TS_{t,k}^TS_{t,k}A)^{-1}A^T(Ax_t-b). \tag{109}$$

Let us decompose b as $b = Ax^* + b^\perp$ and note that $A^Tb^\perp = 0$ which gives us:

$$x_{t+1} = x_t - \mu\frac{1}{q}\sum_{k=1}^{q}(A^TS_{t,k}^TS_{t,k}A)^{-1}A^TAe_t, \tag{110}$$

where $e_t := w_t - w^*$. Subtracting x^* from both sides, we obtain an equation in terms of the error vector e_t only:

$$\begin{aligned} e_{t+1} &= e_t - \mu\frac{1}{q}\sum_{k=1}^{q}(A^TS_{t,k}^TS_{t,k}A)^{-1}A^TAe_t \\ &= \left(I - \mu\frac{1}{q}\sum_{k=1}^{q}(A^TS_{t,k}^TS_{t,k}A)^{-1}A^TA\right)e_t. \end{aligned}$$

Let us multiply both sides by A from the left and define $Q_{t,k} := A(A^TS_{t,k}^TS_{t,k}A)^{-1}A^T$ and we will have the following equation:

$$e_{t+1}^A = \left(I - \mu\frac{1}{q}\sum_{k=1}^{q}Q_{t,k}\right)e_t^A.$$

We now analyze the expectation of ℓ_2 norm of e_{t+1}^A:

$$\begin{aligned}\mathbb{E}[\|e_{t+1}^A\|_2^2] &= \mathbb{E}\left[\left\|\frac{1}{q}\sum_{k=1}^{q}(I-\mu Q_{t,k})e_t^A\right\|_2^2\right]\\ &= \frac{1}{q^2}\,\mathbb{E}\left[\sum_{k=1}^{q}\sum_{l=1}^{q}\langle(I-\mu Q_{t,k})e_t^A,(I-\mu Q_{t,l})e_t^A\rangle\right]\\ &= \frac{1}{q^2}\sum_{k=1}^{q}\sum_{l=1}^{q}\mathbb{E}\left[\langle(I-\mu Q_{t,k})e_t^A,(I-\mu Q_{t,l})e_t^A\rangle\right]. \quad (111)\end{aligned}$$

The contribution for $k\neq l$ in the double summation of (111) is equal to zero because for $k\neq l$, we have

$$\begin{aligned}&\mathbb{E}\left[\langle(I-\mu Q_{t,k})e_t^A,(I-\mu Q_{t,l})e_t^A\rangle\right] =\\ &\quad=\langle\mathbb{E}[(I-\mu Q_{t,k})e_t^A],\mathbb{E}[(I-\mu Q_{t,l})e_t^A]\rangle\\ &\quad=\langle\mathbb{E}[(I-\mu Q_{t,k})e_t^A],\mathbb{E}[(I-\mu Q_{t,k})e_t^A]\rangle\\ &\quad=\left\|\mathbb{E}[(I-\mu Q_{t,k})e_t^A]\right\|_2^2.\end{aligned}$$

The term in the last line above is zero for $\mu=\frac{1}{\theta_1}$:

$$\begin{aligned}\mathbb{E}[(I-\mu Q_{t,k})e_t^A] &= \mathbb{E}[(I-\mu A(A^TS_{t,k}^TS_{t,k}A)^{-1}A^T)e_t^A]\\ &= (I-\mu\theta_1 A(A^TA)^{-1}A^T)e_t^A\\ &= (I-\mu\theta_1 UU^T)e_t^A\\ &= (I-UU^T)e_t^A=0\end{aligned}$$

where we used $A=U\Sigma V^T$. For the rest of the proof, we assume that we set $\mu=1/\theta_1$. Now that we know the contribution from terms with $k\neq l$ is zero, the expansion in (111) can be rewritten as

$$\begin{aligned}\mathbb{E}[\|e_{t+1}^A\|_2^2] &= \frac{1}{q^2}\sum_{k=1}^{q}\mathbb{E}\left[\langle(I-\mu Q_{t,k})e_t^A,(I-\mu Q_{t,k})e_t^A\rangle\right]\\ &= \frac{1}{q^2}\sum_{k=1}^{q}\mathbb{E}[\|(I-\mu Q_{t,k})e_t^A\|_2^2]\\ &= \frac{1}{q}\,\mathbb{E}[\|(I-\mu Q_{t,1})e_t^A\|_2^2]\\ &= \frac{1}{q}\left(\|e_t^A\|_2^2+\mu^2\,\mathbb{E}[\|Q_{t,1}e_t^A\|_2^2]-2\mu(e_t^A)^T\,\mathbb{E}[Q_{t,1}]e_t^A\right)\\ &= \frac{1}{q}\left(\mu^2\,\mathbb{E}[\|Q_{t,1}e_t^A\|_2^2]-\|e_t^A\|_2^2\right)\\ &= \frac{1}{q}\left(\mu^2(e_t^A)^T\,\mathbb{E}[Q_{t,1}^TQ_{t,1}]e_t^A-\|e_t^A\|_2^2\right).\end{aligned}$$

The term $\mathbb{E}[Q_{t,1}^TQ_{t,1}]$ can be simplified using SVD decomposition $A=U\Sigma V^T$. This gives us $Q_{t,k}=U(U^TS_{t,k}^TS_{t,k}U)^{-1}U^T$ and furthermore we have:

$$\begin{aligned}&\mathbb{E}[Q_{t,1}^TQ_{t,1}] =\\ &\quad=\mathbb{E}[U(U^TS_{t,1}^TS_{t,1}U)^{-1}U^TU(U^TS_{t,1}^TS_{t,1}U)^{-1}U^T]\\ &\quad=\mathbb{E}[U(U^TS_{t,1}^TS_{t,1}U)^{-1}(U^TS_{t,1}^TS_{t,1}U)^{-1}U^T]\\ &\quad=U\,\mathbb{E}[(U^TS_{t,1}^TS_{t,1}U)^{-2}]U^T\\ &\quad=\theta_2UU^T.\end{aligned}$$

Plugging this in, we obtain:

$$\begin{aligned}\mathbb{E}[\|e_{t+1}^A\|_2^2] &= \frac{1}{q}\left(\theta_2\mu^2(e_t^A)^TUU^Te_t^A-\|e_t^A\|_2^2\right)\\ &= \frac{1}{q}\left(\theta_2\mu^2\|e_t^A\|_2^2-\|e_t^A\|_2^2\right)\\ &= \frac{\theta_2\mu^2-1}{q}\|e_t^A\|_2^2\\ &= \frac{1}{q}\left(\frac{\theta_2}{\theta_1^2}-1\right)\|e_t^A\|_2^2.\end{aligned}$$

■

Proof of Corollary 3.3: Taking the expectation with respect to the sketching matrices $S_{t,k}$, $k=1,\ldots,q$ of both sides of the equation given in Theorem 3.2, we obtain

$$\mathbb{E}[\|e_{t+1}^A\|_2^2]=\frac{1}{q}\left(\frac{\theta_2}{\theta_1^2}-1\right)\mathbb{E}[\|e_t^A\|_2^2].$$

This gives us the relationship between the initial error (when we initialize x_0 to be the zero vector) and the expected error in iteration t:

$$\mathbb{E}[\|e_t^A\|_2^2]=\frac{1}{q^t}\left(\frac{\theta_2}{\theta_1^2}-1\right)^t\|Ax^*\|_2^2.$$

It follows that the expected error reaches ϵ-accuracy with respect to the initial error at iteration T where:

$$\begin{aligned}\frac{1}{q^T}\left(\frac{\theta_2}{\theta_1^2}-1\right)^T &= \epsilon\\ q^T\left(\frac{\theta_2}{\theta_1^2}-1\right)^{-T} &= \frac{1}{\epsilon}\\ T\left(\log(q)-\log\left(\frac{\theta_2}{\theta_1^2}-1\right)\right) &= \log(1/\epsilon)\\ T &= \frac{\log(1/\epsilon)}{\log(q)-\log\left(\frac{\theta_2}{\theta_1^2}-1\right)}.\end{aligned}$$

Each iteration requires communicating a d-dimensional vector for every worker, and we have q workers and the algorithm runs for T iterations, hence the communication load is Tqd.

The computational load per worker node at each iteration is as follows:

- Sketching A: mnd multiplications
- Computing $\tilde{H}_{t,k}$: md^2 multiplications
- Computing g_t: $O(nd)$ operations
- Solving $\tilde{H}_{t,k}^{-1}g_t$: $O(d^3)$ operations.

■

Lemma B.2 ([4]): For the Gaussian sketch matrix $S\in\mathbb{R}^{m\times n}$ with i.i.d. entries distributed as $\mathcal{N}(0,1/\sqrt{m})$ where $m\geq d$, and for $U\in\mathbb{R}^{n\times d}$ with $U^TU=I_d$, the following are true:

$$\begin{aligned}\mathbb{E}[(U^TS^TSU)^{-1}] &= \theta_1 I_d,\\ \mathbb{E}[(U^TS^TSU)^{-2}] &= \theta_2 I_d, \quad (112)\end{aligned}$$

where θ_1 and θ_2 are defined as

$$\begin{aligned}\theta_1 &\coloneqq \frac{m}{m-d-1},\\ \theta_2 &\coloneqq \frac{m^2(m-1)}{(m-d)(m-d-1)(m-d-3)}. \quad (113)\end{aligned}$$

Proof of Theorem 3.5: The optimal update direction is given by

$$\Delta_t^* = ((H_t^{1/2})^T H_t^{1/2})^{-1} g_t = H_t^{-1} g_t$$

and the estimate update direction due to a single sketch is given by

$$\hat{\Delta}_{t,k} = \alpha_s ((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2})^{-1} g_t.$$

where $\alpha_s \in \mathbb{R}$ is the step size scaling factor to be determined. Letting $S_{t,k}$ be a Gaussian sketch, the bias can be written as

$$\begin{aligned}
&\mathbb{E}[H_t^{1/2}(\hat{\Delta}_{t,k} - \Delta_t^*)] = \\
&= \mathbb{E}[\alpha_s H_t^{1/2} ((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2})^{-1} g_t - H_t^{1/2} H_t^{-1} g_t] \\
&= \alpha_s H_t^{1/2}\, \mathbb{E}[((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2})^{-1}] g_t - H_t^{1/2} H_t^{-1} g_t \\
&= \alpha_s \theta_1 H_t^{1/2} ((H_t^{1/2})^T H_t^{1/2})^{-1} g_t - H_t^{1/2} H_t^{-1} g_t \\
&= (\alpha_s \theta_1 - 1)\, H_t^{1/2} H_t^{-1} g_t\,.
\end{aligned}$$

In the third line, we plug in the mean of $((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2})^{-1}$ which is distributed as inverse Wishart distribution (see Lemma B.2). This calculation shows that the single sketch estimator gives an unbiased update direction for $\alpha_s = 1/\theta_1$.

The variance analysis is as follows:

$$\begin{aligned}
&\mathbb{E}[\|H_t^{1/2}(\hat{\Delta}_{t,k} - \Delta_t^*)\|_2^2] = \\
&= \mathbb{E}[\hat{\Delta}_{t,k}^T H_t \hat{\Delta}_{t,k} + \Delta_t^{*T} H_t \Delta_t^* - 2\Delta_t^{*T} H_t \hat{\Delta}_{t,k}] \\
&= \alpha_s^2 g_t^T\, \mathbb{E}[((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2})^{-1} H_t \\
&\qquad ((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2})^{-1}] g_t + (1 - 2\alpha_s\theta_1)\, g_t^T H_t^{-1} g_t.
\end{aligned}$$

Plugging $H_t^{1/2} = U\Sigma V^T$ into the first term and assuming $H_t^{1/2}$ has full column rank, the expectation term becomes

$$\begin{aligned}
&\mathbb{E}[((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2})^{-1} H_t ((H_t^{1/2})^T S_{t,k}^T S_{t,k} H_t^{1/2})^{-1}] = \\
&= V\Sigma^{-1}\, \mathbb{E}[(U^T S_{t,k}^T S_{t,k} U)^{-2}] \Sigma^{-1} V^T \\
&= V\Sigma^{-1} (\theta_2 I_d) \Sigma^{-1} V^T \\
&= \theta_2 V \Sigma^{-2} V^T,
\end{aligned}$$

where the third line follows due to Lemma B.2. Because $H_t^{-1} = V\Sigma^{-2}V^T$, the variance becomes:

$$\begin{aligned}
&\mathbb{E}[\|H_t^{1/2}(\hat{\Delta}_{t,k} - \Delta_t^*)\|_2^2] = \\
&\quad = \left(\alpha_s^2\theta_2 + 1 - 2\alpha_s\theta_1\right) g_t^T V \Sigma^{-2} V^T g_t \\
&\quad = \left(\alpha_s^2\theta_2 + 1 - 2\alpha_s\theta_1\right) \|\Sigma^{-1} V^T g_t\|_2^2\,.
\end{aligned}$$

It follows that the variance is minimized when α_s is chosen as $\alpha_s = \theta_1/\theta_2$. ∎

C. Proofs of Theorems and Lemmas in Section IV

Proof of Lemma 4.2: In the following, we assume that we are in the regime where n approaches infinity.

The expectation term $\mathbb{E}[(U^T S^T SU + \lambda_2 I)^{-1}]$ is equal to the identity matrix times a scalar (i.e. cI_d) because it is signed permutation invariant, which we show as follows. Let $P \in \mathbb{R}^{d\times d}$ be a permutation matrix and $D \in \mathbb{R}^{d\times d}$ be an invertible diagonal sign matrix (-1 and $+1$'s on the diagonals). A matrix M is signed permutation invariant if $(DP)M(DP)^T = M$. We note that the signed permutation matrix is orthogonal: $(DP)^T(DP) = P^T D^T DP = P^T P = I_d$, which we later use in the sequel.

$$\begin{aligned}
&(DP)\, \mathbb{E}_S[(U^T S^T SU + \lambda_2 I)^{-1}](DP)^T = \\
&= \mathbb{E}_S[(DP)(U^T S^T SU + \lambda_2 I)^{-1}(DP)^T] \\
&= \mathbb{E}_S[((DP)^T U^T S^T SU(DP) + \lambda_2 I)^{-1}] \\
&= \mathbb{E}_{SUPD}[\mathbb{E}_S[((DP)^T U^T S^T SU(DP) + \lambda_2 I)^{-1} | SUPD]] \\
&= \mathbb{E}_{SUPD}[((DP)^T U^T S^T SU(DP) + \lambda_2 I)^{-1}] \\
&= \mathbb{E}_{SU'}[(U'^T S^T SU' + \lambda_2 I)^{-1}]
\end{aligned}$$

where we made the variable change $U' = UDP$ and note that U' has orthonormal columns because DP is an orthogonal transformation. $SUPD$ and SU have the same distribution because PD is an orthogonal transformation and S is a Gaussian matrix. This shows that $\mathbb{E}[(U^T S^T SU + \lambda_2 I)^{-1}]$ is signed permutation invariant.

Now that we established that $\mathbb{E}[(U^T S^T SU + \lambda_2 I)^{-1}]$ is equal to the identity matrix times a scalar, we move on to find the value of the scalar. We use the identity $\mathbb{E}_{DP}[(DP)Q(DP)^T] = \frac{\operatorname{tr} Q}{d} I_d$ for $Q \in \mathbb{R}^{d\times d}$ where the diagonal entries of D are sampled from the Rademacher distribution and P is sampled uniformly from the set of all possible permutation matrices. We already established that $\mathbb{E}[(U^T S^T SU + \lambda_2 I)^{-1}]$ is equal to $(DP)\, \mathbb{E}_S[(U^T S^T SU + \lambda_2 I)^{-1}](DP)^T$ for any signed permutation matrix of the form DP. It follows that

$$\begin{aligned}
&\mathbb{E}[(U^T S^T SU + \lambda_2 I)^{-1}] = \\
&\quad = (DP)\, \mathbb{E}_S[(U^T S^T SU + \lambda_2 I)^{-1}](DP)^T \\
&\quad = \frac{1}{|R|} \sum_{DP \in R} (DP)\, \mathbb{E}_S[(U^T S^T SU + \lambda_2 I)^{-1}](DP)^T \\
&\quad = \mathbb{E}_{DP}[(DP)\, \mathbb{E}_S[(U^T S^T SU + \lambda_2 I)^{-1}](DP)^T] \\
&\quad = \frac{1}{d} \operatorname{tr}(\mathbb{E}_S[(U^T S^T SU + \lambda_2 I)^{-1}]) I_d
\end{aligned}$$

where we define R to be the set of all possible signed permutation matrices DP in going from line 1 to line 2.

By Lemma 4.1, the trace term is equal to $d \times \theta_3(d/m, \lambda_2)$, which concludes the proof. ∎

Proof of Theorem 4.3: Closed form expressions for the optimal solution and the output of the k'th worker are as follows:

$$\begin{aligned}
x^* &= (A^T A + \lambda_1 I_d)^{-1} A^T b, \\
\hat{x}_k &= (A^T S_k^T S_k A + \lambda_2 I_d)^{-1} A^T S_k^T S_k b.
\end{aligned}$$

Equivalently, x^* can be written as:

$$x^* = \arg\min \left\| \begin{bmatrix} A \\ \sqrt{\lambda_1} I_d \end{bmatrix} x - \begin{bmatrix} b \\ 0_d \end{bmatrix} \right\|_2^2.$$

This allows us to decompose $\begin{bmatrix} b \\ 0_d \end{bmatrix}$ as

$$\begin{bmatrix} b \\ 0_d \end{bmatrix} = \begin{bmatrix} A \\ \sqrt{\lambda_1} I_d \end{bmatrix} x^* + b^\perp$$

where $b^\perp = \begin{bmatrix} b_1^\perp \\ b_2^\perp \end{bmatrix}$ with $b_1^\perp \in \mathbb{R}^n$ and $b_2^\perp \in \mathbb{R}^d$. From the above equation we obtain $b_2^\perp = -\sqrt{\lambda_1} x^*$ and $\begin{bmatrix} A^T & \sqrt{\lambda_1} I_d \end{bmatrix} b^\perp = A^T b_1^\perp + \sqrt{\lambda_1} b_2^\perp = 0$.

The bias of $\hat{x}_k$ is given by (omitting the subscript k in S_k for simplicity)

$$\begin{aligned}
&\mathbb{E}[A(\hat{x}_k - x^*)] = \\
&= \mathbb{E}[A(A^T S^T S A + \lambda_2 I_d)^{-1} A^T S^T S b - A x^*] \\
&= \mathbb{E}[U(U^T S^T S U + \lambda_2 \Sigma^{-2})^{-1} U^T S^T S (A x^* + b_1^\perp)] - A x^* \\
&= \mathbb{E}[-\lambda_2 U (U^T S^T S U + \lambda_2 \Sigma^{-2})^{-1} \Sigma^{-1} V^T x^*] \\
&\quad + \mathbb{E}[U(U^T S^T S U + \lambda_2 \Sigma^{-2})^{-1} U^T S^T S b_1^\perp] \,.
\end{aligned}$$

By the assumption $\Sigma = \sigma I_d$, the bias becomes

$$\begin{aligned}
&\mathbb{E}[A(\hat{x}_k - x^*)] = \\
&\quad = \mathbb{E}[-\lambda_2 \sigma^{-1} U (U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1} V^T x^*] \\
&\qquad + \mathbb{E}[U(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1} U^T S^T S b_1^\perp] \,. \qquad (114)
\end{aligned}$$

The first expectation term of (114) can be evaluated using Lemma 4.2 (as n goes to infinity):

$$\begin{aligned}
&\mathbb{E}[-\lambda_2 \sigma^{-1} U (U^T S^T S U + \lambda_2 I_d)^{-1} V^T x^*] = \\
&\qquad = -\lambda_2 \sigma^{-1} \theta_3(d/m, \lambda_2 \sigma^{-2}) U V^T x^* . \qquad (115)
\end{aligned}$$

To find the second expectation term in (114), let us first consider the full SVD of A given by $A = \begin{bmatrix} U & U^\perp \end{bmatrix} \begin{bmatrix} \Sigma \\ 0_{(n-d)\times d} \end{bmatrix} V^T$ where $U \in \mathbb{R}^{n \times d}$ and $U^\perp \in \mathbb{R}^{n \times (n-d)}$. The matrix $\begin{bmatrix} U & U^\perp \end{bmatrix}$ is an orthogonal matrix, which implies $UU^T + U^\perp (U^\perp)^T = I_d$. If we insert $UU^T + U^\perp (U^\perp)^T = I_d$ between S and $b_1^\perp$, the second term of (114) becomes

$$\begin{aligned}
&\mathbb{E}[U(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1} U^T S^T S b_1^\perp] = \\
&= \mathbb{E}[U(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1} U^T S^T S U U^T b_1^\perp] \\
&\quad + \mathbb{E}[U(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1} U^T S^T S U^\perp (U^\perp)^T b_1^\perp] \\
&= \mathbb{E}[U(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1} U^T S^T S U U^T b_1^\perp] \\
&= U(I_d - \lambda_2 \sigma^{-2} \mathbb{E}[(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1}]) U^T b_1^\perp \\
&= (1 - \lambda_2 \sigma^{-2} \theta_3(d/m, \lambda_2 \sigma^{-2})) U U^T b_1^\perp \,.
\end{aligned}$$

In these derivations, we have used the identity $\mathbb{E}_S[U(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1} U^T S^T S U^\perp (U^\perp)^T b_1^\perp] = \mathbb{E}_{SU}[\mathbb{E}_S[U(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1} U^T S^T S U^\perp (U^\perp)^T b_1^\perp | SU]]$, which is equal to 0. This follows from $\mathbb{E}_S[SU^\perp | SU] = 0$ as U and $U^\perp$ are orthogonal. The last line follows from Lemma 4.2, as n goes to infinity.

We note that $U^T b_1^\perp = \lambda_1 \Sigma^{-1} V^T x^*$ and for $\Sigma = \sigma I_d$, this becomes $U^T b_1^\perp = \lambda_1 \sigma^{-1} V^T x^*$. Bringing the pieces together, we have the bias equal to (as n goes to infinity):

$$\begin{aligned}
\mathbb{E}[A(\hat{x}_k - x^*)] &= -\lambda_2 \sigma^{-1} \theta_3(d/m, \lambda_2 \sigma^{-2}) U V^T x^* \\
&\quad + \lambda_1 \sigma^{-1} (1 - \lambda_2 \sigma^{-2} \theta_3(d/m, \lambda_2 \sigma^{-2})) U V^T x^* \\
&= \sigma^{-1} (\lambda_1 - \lambda_2 \theta_3(d/m, \lambda_2 \sigma^{-2})(1 + \lambda_1 \sigma^{-2})) U V^T x^* .
\end{aligned}$$

If there is a value of $\lambda_2 > 0$ that satisfies $\lambda_1 - \lambda_2 \theta_3(d/m, \lambda_2 \sigma^{-2})(1 + \lambda_1 \sigma^{-2}) = 0$, then that value of λ_2 makes $\hat{x}_k$ an unbiased estimator. Equivalently,

$$\begin{aligned}
&-\lambda_2 \sigma^{-2} + \frac{d}{m} - 1 + \sqrt{(-\lambda_2 \sigma^{-2} + \frac{d}{m} - 1)^2 + 4 \lambda_2 \sigma^{-2} \frac{d}{m}} = \\
&= 2 \frac{d}{m \sigma^2} \frac{\lambda_1}{1 + \lambda_1 \sigma^{-2}} , \qquad (116)
\end{aligned}$$

where we note that the LHS is a monotonically increasing function of λ_2 in the regime $\lambda_2 \geq 0$ and it attains its minimum in this regime at $\lambda_2 = 0$. Analyzing this equation using these observations, for the cases of $m > d$ and $m \leq d$ separately, we find that for the case of $m \leq d$, we need the following to be satisfied for zero bias:

$$\begin{aligned}
2 \frac{d}{m \sigma^2} \frac{\lambda_1}{1 + \lambda_1 / \sigma^2} \geq 2 \left(\frac{d}{m} - 1 \right) , \text{ or more simply,} \\
\lambda_1 \geq \sigma^2 \left(\frac{d}{m} - 1 \right) ,
\end{aligned}$$

whereas there is no additional condition on λ_1 for the case of $m > d$.

The value of λ_2 that will lead to zero bias can be computed by solving the equation (116) where the expression for the inverse of the left-hand side is given by $LHS^{-1}(y) = \frac{y \sigma^{-2} d/m - d/m + 1}{y^{-1} - \sigma^{-2}}$. We evaluate the inverse at $y = \lambda_1 / (1 + \lambda_1 / \sigma^2)$ and obtain the following expression for λ_2^*:

$$\lambda_2^* = \lambda_1 - \frac{d}{m} \frac{\lambda_1}{1 + \lambda_1 / \sigma^2} .$$

■

Proof of Theorem 4.5: In the following, we omit the subscripts in $S_{t,k}$ for simplicity. Using the SVD decomposition of $H_t^{1/2} = U \Sigma V^T$, the bias can be written as

$$\begin{aligned}
&\mathbb{E}[H_t^{1/2} (\hat{\Delta}_{t,k} - \Delta_t^*)] = \\
&= U \, \mathbb{E}[(U^T S^T S U + \lambda_2 \Sigma^{-2})^{-1}] \Sigma^{-1} V^T g_t \\
&\quad - U (I_d + \lambda_1 \Sigma^{-2})^{-1} \Sigma^{-1} V^T g_t \,.
\end{aligned}$$

By the assumption that $\Sigma = \sigma I_d$, the bias term can be simplified as

$$\begin{aligned}
&\mathbb{E}[H_t^{1/2} (\hat{\Delta}_{t,k} - \Delta_t^*)] = \\
&= \sigma^{-1} U \, \mathbb{E}[(U^T S^T S U + \lambda_2 \sigma^{-2} I_d)^{-1}] V^T g_t \\
&\quad - \sigma^{-1} (1 + \lambda_1 \sigma^{-2})^{-1} U V^T g_t .
\end{aligned}$$

By Lemma 4.2, as n goes to infinity, we have

$$\begin{aligned}
&\mathbb{E}[H_t^{1/2} (\hat{\Delta}_{t,k} - \Delta_t^*)] = \\
&= \sigma^{-1} \left(\theta_3(d/m, \lambda_2 \sigma^{-2}) - \frac{1}{1 + \lambda_1 \sigma^{-2}} \right) U V^T g_t \\
&= \frac{-\frac{\lambda_2}{\sigma^2} + \frac{d}{m} - 1 + \sqrt{(-\frac{\lambda_2}{\sigma^2} + \frac{d}{m} - 1)^2 + 4 \frac{\lambda_2}{\sigma^2} \frac{d}{m}}}{2 \lambda_2 \sigma^{-1} d/m} U V^T g_t \\
&\quad - \frac{\sigma^{-1}}{1 + \lambda_1 \sigma^{-2}} U V^T g_t .
\end{aligned}$$

The bias becomes zero for the value of λ_2 that satisfies the following equation:

$$\sqrt{\left(-\sigma^{-2}+\frac{1}{\lambda_2}\left(\frac{d}{m}-1\right)\right)^2+4\sigma^{-2}\frac{d}{m\lambda_2}}-\sigma^{-2} + \frac{1}{\lambda_2}\left(\frac{d}{m}-1\right)=2\sigma^{-2}\frac{d}{m}\frac{1}{1+\lambda_1\sigma^{-2}}. \quad (117)$$

In the regime where $\lambda_2 \geq 0$, the LHS of (117) is always non-negative and is monotonically decreasing in λ_2. The LHS approaches zero as $\lambda_2 \to \infty$. We now consider the following cases:

- Case 1: $m \leq d$. Because $d/m - 1 \geq 0$, as $\lambda_2 \to 0$, the LHS goes to infinity. Since the LHS can take any values between 0 and ∞, there is an appropriate λ_2^* value that makes the bias zero for any λ_1.
- Case 2: $m > d$. In this case, $d/m - 1 < 0$. The maximum of LHS in this case is reached as $\lambda_2 \to 0$ and it is equal to $2\sigma^{-2}\frac{d}{m-d}$. As long as $2\sigma^{-2}\frac{d}{m}\frac{1}{1+\lambda_1\sigma^{-2}} \leq 2\sigma^{-2}\frac{d}{m-d}$ is true, then we can drive the bias down to zero. More simply, this corresponds to $\lambda_1\sigma^{-2} \geq -d/m$, which is always true. Therefore in the case of $m > d$ as well, there is a λ_2^* value for any value of λ_1 that will drive the bias down to zero.

To sum up, for any given value for the regularization parameter λ_1, it is possible to find a λ_2^* value to make the sketched update direction unbiased. The optimal value for λ_2 is given by $LHS^{-1}(2\sigma^{-2}\frac{d}{m}\frac{1}{1+\lambda_1\sigma^{-2}})$ where $LHS^{-1}(y) = \frac{4\sigma^{-2}y^{-1}d/m+2(d/m-1)}{y+2\sigma^{-2}}$, which simplifies to the following expression:

$$\lambda_2^* = \left(\lambda_1+\frac{d}{m}\sigma^2\right)\left(1-\frac{d/m}{1+\lambda_1\sigma^{-2}+d/m}\right).$$

■

D. Proofs of Theorems and Lemmas in Section V

The proof of the privacy result mainly follows due to Theorem B.3 (stated below for completeness) which is a result from [37].

Proof of Theorem 5.2: For some ε, δ and matrix A_c, if there exist values for m such that the smallest singular value of A_c satisfies $\sigma_{\min}(A_c) \geq w$, then using Theorem B.3, we find that the sketch size m has to satisfy the following for (ε, δ)-differential privacy:

$$m \leq \frac{1}{8\ln(4/\delta)}\left(\left(\frac{\sigma_{\min}^2}{B^2}-1\right)\frac{1}{\frac{1}{\varepsilon}+\frac{1}{\ln(4/\delta)}}-2\ln(4/\delta)\right)^2 = \frac{1}{8\beta}\left(\left(\frac{\sigma_{\min}^2}{B^2}-1\right)\frac{\varepsilon\beta}{\varepsilon+\beta}-2\beta\right)^2, \quad (118)$$

where we have set $\delta = 4/e^\beta$ in the second line. For the first line to follow from Theorem B.3, we also need the condition $\frac{\sigma_{\min}^2}{B^2} \geq 3 + 2\frac{\beta}{\varepsilon}$ to be satisfied. Note that the rows of A_c have bounded ℓ_2-norm of $B_0\sqrt{d+1}$. We now substitute $B = B_0\sqrt{d+1}$ and $\sigma_{\min} = \sigma_0\sqrt{n}$ to obtain the simplified condition

$$\frac{n}{d+1} \geq \left(3+2\frac{\beta}{\varepsilon}\right)\frac{B_0^2}{\sigma_0^2},$$

where B_0 and σ_0 are constants. Assuming this condition is satisfied, then we pick the sketch size m as (118) which can also be simplified:

$$m = O\left(\beta\frac{n^2}{(d+1)^2}\frac{\varepsilon^2}{(\varepsilon+\beta)^2}\right).$$

Note that the above arguments are for the privacy of a single sketch (i.e., S_kA_c). In the distributed setting where the adversary can attack all of the sketched data $S_1A_c, \ldots, S_qA_c$, we can consider all of the sketched data to be a single sketch with size mq. Based on this argument, we can pick the sketch size as

$$m = O\left(\frac{\beta}{q}\frac{n^2}{(d+1)^2}\frac{\varepsilon^2}{(\varepsilon+\beta)^2}\right).$$

■

Theorem B.3 (Differential privacy for random projections [37]): Fix $\varepsilon > 0$ and $\delta \in (0, 1/e)$. Fix $B > 0$. Fix a positive integer m and let w be such that

$$w^2 = B^2\left(1+\frac{1+\frac{\varepsilon}{\ln(4/\delta)}}{\varepsilon}\left(2\sqrt{2m\ln(4/\delta)}+2\ln(4/\delta)\right)\right). \quad (119)$$

Let A be an $(n \times d)$-matrix with $d < m$ and where each row of A has bounded ℓ_2-norm of B. Given that $\sigma_{\min}(A) \geq w$, the algorithm that picks an $(m \times n)$-matrix R whose entries are iid samples from the normal distribution $\mathcal{N}(0,1)$ and publishes the projection RA is (ε, δ)-differentially private.

E. Proofs of Theorems and Lemmas in Section A

Proof of Lemma A.1: The expectation of the difference between the costs $f(\bar{x})$ and $f(x^*)$ is given by

$$\begin{aligned}\mathbb{E}[f(\bar{x})] - f(x^*) &= \mathbb{E}[\|A(\bar{x}-x^*)+Ax^*-b\|_2^2] - f(x^*)\\ &= \mathbb{E}[\|A(\bar{x}-x^*)\|_2^2 + \|Ax^*-b\|_2^2] - f(x^*)\\ &= \mathbb{E}[\|A(\bar{x}-x^*)\|_2^2], \end{aligned} \quad (120)$$

where we have used the orthogonality property of the optimal least squares solution x^* given by the normal equations $A^T(Ax^*-b)=0$. Next, we have

$$\begin{aligned}&\mathbb{E}[\|A(\bar{x}-x^*)\|_2^2] = \\ &= \mathbb{E}\left[\left\|\frac{1}{q}\sum_{k=1}^q(A\hat{x}_k - Ax^*)\right\|_2^2\right]\\ &= \frac{1}{q^2}\mathbb{E}\left[\sum_{k=1}^q\sum_{l=1}^q\langle A\hat{x}_k - Ax^*, A\hat{x}_l - Ax^*\rangle\right]\\ &= \frac{1}{q^2}\sum_{k=1}^q\mathbb{E}\left[\|A\hat{x}_k - Ax^*\|_2^2\right]\\ &\quad + \frac{1}{q^2}\sum_{k\neq l\,,1\leq k,l\leq q}\mathbb{E}\left[\langle A\hat{x}_k - Ax^*, A\hat{x}_l - Ax^*\rangle\right]\\ &= \frac{1}{q}\mathbb{E}\left[\|A\hat{x}-Ax^*\|_2^2\right] + \frac{q-1}{q}\|\,\mathbb{E}[A\hat{x}]-Ax^*\|_2^2. \end{aligned}$$

■

Proof of Lemma A.2: The bias of the single sketch estimator can be expanded as follows:

$$\begin{aligned}
&\| \mathbb{E}[A\hat{x}] - Ax^*\|_2 = \\
&= \| \mathbb{E}[A(A^TS^TSA)^{-1}A^TS^TS(Ax^* + b^\perp)] - Ax^*\|_2 \\
&= \|U\, \mathbb{E}[(U^TS^TSU)^{-1}U^TS^TSb^\perp\|_2 \\
&= \| \mathbb{E}[(U^TS^TSU)^{-1}U^TS^TSb^\perp\|_2 \\
&= \| \mathbb{E}[Qz]\|_2,
\end{aligned}$$

where we define $Q := (U^TS^TSU)^{-1}$ and $z := U^TS^TSb^\perp$. The term $\| \mathbb{E}[Qz]\|_2^2$ can be upper bounded as follows when conditioned on the event E:

$$\begin{aligned}
&\| \mathbb{E}[Qz]\|_2^2 = \mathbb{E}[Qz]^T\, \mathbb{E}[Qz] = \mathbb{E}_S[Qz]^T\, \mathbb{E}_{S'}[Q'z'] \\
&= \mathbb{E}_{S_k}\, \mathbb{E}_{S_k'}[z^TQQ'z'] \\
&= \frac{1}{2}\, \mathbb{E}_S\, \mathbb{E}_{S'}[(z+z')^TQQ'(z+z') - z^TQQ'z - z'^TQQ'z'] \\
&\leq \frac{1}{2}\, \mathbb{E}_S\, \mathbb{E}_{S'}[\|z+z'\|_2^2(1+\epsilon)^2 - (\|z\|_2^2 + \|z'\|_2^2)(1-\epsilon)^2] \\
&= \mathbb{E}_S\, \mathbb{E}_{S'}[\big(\|z\|_2^2 2\epsilon + \|z'\|_2^2 2\epsilon + z^Tz'(1+\epsilon)^2\big)] \\
&= 4\epsilon\, \mathbb{E}[\|z\|_2^2] + (1+\epsilon)^2 \| \mathbb{E}[z]\|_2^2\,,
\end{aligned}$$

where the inequality follows from the inequality $(1-\epsilon)I_d \preceq Q \preceq (1+\epsilon)I_d$ and some simple bounds for the minimum and maximum eigenvalues of the product of two positive definite matrices. Furthermore, the expectation of z is equal to zero because $\mathbb{E}[z] = \mathbb{E}[U^TS^TSb^\perp] = U^T\, \mathbb{E}[S^TS]b^\perp = U^Tb^\perp = 0$. Hence we obtain the claimed bound $\| \mathbb{E}[A\hat{x}|E] - Ax^*\|_2 \leq \sqrt{4\epsilon\, \mathbb{E}[\|z\|_2^2|E]}$. ∎

Proof of Lemma A.3: For the randomized Hadamard sketch (ROS), the term $\mathbb{E}[\|z\|_2^2]$ can be expanded as follows. We will assume that all the expectations are conditioned on the event E, which we defined earlier as $(1-\epsilon)I_d \preceq Q \preceq (1+\epsilon)I_d$.

$$\begin{aligned}
\mathbb{E}[\|z\|_2^2] &= \mathbb{E}\left[b^{\perp T}\frac{1}{m}\sum_{i=1}^m s_is_i^TUU^T\frac{1}{m}\sum_{j=1}^m s_js_j^Tb^\perp\right] \\
&= \mathbb{E}\left[\frac{1}{m^2}\sum_{i=1}^m\sum_{j=1}^m b^{\perp T}s_is_i^TUU^Ts_js_j^Tb^\perp\right] \\
&= \frac{1}{m^2}\sum_{1\leq i=j\leq m} b^{\perp T}\, \mathbb{E}\left[s_is_i^TUU^Ts_js_j^T\right]b^\perp \\
&\quad + \frac{1}{m^2}\sum_{i\neq j,\,1\leq i,j\leq m} b^{\perp T}\, \mathbb{E}[s_is_i^T]UU^T\, \mathbb{E}[s_js_j^T]b^\perp \\
&= \frac{m}{m^2}\, b^{\perp T}\, \mathbb{E}\left[s_1s_1^TUU^Ts_1s_1^T\right]b^\perp \\
&\quad + \frac{1}{m^2}\sum_{i\neq j,\,1\leq i,j\leq m} b^{\perp T}I_nUU^TI_nb^\perp \\
&= \frac{1}{m}\, b^{\perp T}\, \mathbb{E}\left[s_1s_1^TUU^Ts_1s_1^T\right]b^\perp,
\end{aligned}$$

where we have used the independence of s_i and s_j, $i \neq j$. This is true because of the assumption that the matrix P corresponds to sampling with replacement.

$$\begin{aligned}
b^{\perp T}\, \mathbb{E}\left[s_1s_1^TUU^Ts_1s_1^T\right]b^\perp &= \\
&= \mathbb{E}[(s_1^TUU^Ts_1)(s_1^Tb^\perp b^{\perp T}s_1)] \\
&= \mathbb{E}[(s_1^TUU^Ts_1)(b^{\perp T}s_1)^2] \\
&= \frac{1}{n}\sum_{i=1}^n \mathbb{E}[(h_i^TDUU^TDh_i)(b^{\perp T}Dh_i)^2],
\end{aligned}$$

where the row vector h_i^T corresponds to the i'th row of the Hadamard matrix H. We also note that the expectation in the last line is with respect to the randomness of D.

Let us define r to be the column vector containing the diagonal entries of the diagonal matrix D, that is, $r := [D_{11}, D_{22}, \ldots, D_{nn}]^T$. Then, the vector Dh_i is equivalent to $\operatorname{diag}(h_i)r$ where $\operatorname{diag}(h_i)$ is the diagonal matrix with the entries of h_i on its diagonal.

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n \mathbb{E}[(h_i^TDUU^TDh_i)(b^{\perp T}Dh_i)^2] = \\
&= \frac{1}{n}\sum_{i=1}^n \mathbb{E}[(r^T\operatorname{diag}(h_i)UU^T\operatorname{diag}(h_i)r)(b^{\perp T}\operatorname{diag}(h_i)r)^2] \\
&= \frac{1}{n}\sum_{i=1}^n \mathbb{E}[b^{\perp T}\operatorname{diag}(h_i)r(r^TPr)r^T\operatorname{diag}(h_i)b^\perp] \\
&= \frac{1}{n}\sum_{i=1}^n b^{\perp T}\operatorname{diag}(h_i)\, \mathbb{E}[r(r^TPr)r^T]\operatorname{diag}(h_i)b^\perp,
\end{aligned}$$

where we have defined $P := \operatorname{diag}(h_i)UU^T\operatorname{diag}(h_i)$. It follows that $\mathbb{E}[r(r^TPr)r^T] = 2P - 2\operatorname{diag}(P) + \operatorname{tr}(P)I_n$. Here, $\operatorname{diag}(P)$ is used to refer to the diagonal matrix with the diagonal entries of P as its diagonal. The trace of P can be easily computed using the cyclic property of matrix trace as $\operatorname{tr}(P) = \operatorname{tr}(\operatorname{diag}(h_i)UU^T\operatorname{diag}(h_i)) = \operatorname{tr}(U^T\operatorname{diag}(h_i)\operatorname{diag}(h_i)U) = \operatorname{tr}(U^TU) = \operatorname{tr}(I_d) = d$. Next, we note that the term $\operatorname{diag}(P)$ can be simplified as $\operatorname{diag}(P)_{jj} = \|\tilde{u}_j\|_2^2$ where $\tilde{u}_j^T$ is the j'th row of U. This leads to

$$\begin{aligned}
b^{\perp T}\operatorname{diag}(P)b^\perp &= \sum_{j=1}^n (b_j^\perp)^2\|\tilde{u}_j\|_2^2 \\
&\geq \sum_{j=1}^n (b_j^\perp)^2 \min_i \|\tilde{u}_i\|_2^2 = \|b^\perp\|_2^2 \min_i \|\tilde{u}_i\|_2^2\,.
\end{aligned}$$

Going back to $\mathbb{E}[\|z\|_2^2]$, we have

$$\begin{aligned}
&\mathbb{E}[\|z\|_2^2] = \\
&= \frac{1}{mn}b^{\perp T}n\operatorname{diag}(h_i)\left(2P - 2\operatorname{diag}(P) + \operatorname{tr}(P)I_n\right)\operatorname{diag}(h_i)b^\perp \\
&= \frac{d}{m}\|b^\perp\|_2^2 - \frac{2}{m}b^{\perp T}\operatorname{diag}(P)b^\perp \\
&\leq \frac{d}{m}\|b^\perp\|_2^2 - \frac{2}{m}\|b^\perp\|_2^2 \min_i \|\tilde{u}_i\|_2^2 \\
&= \frac{1}{m}\|b^\perp\|_2^2(d - 2\min_i \|\tilde{u}_i\|_2^2) \\
&= \frac{d}{m}\left(1 - \frac{2\min_i \|\tilde{u}_i\|_2^2}{d}\right)f(x^*).
\end{aligned}$$

∎

Proof of Lemma A.4: We will assume that all the expectations are conditioned on the event E, which we defined earlier as $(1-\epsilon)I_d \preceq Q \preceq (1+\epsilon)I_d$. For uniform sampling with replacement, we have

$$\begin{aligned}
\mathbb{E}[\|z\|_2^2] &= \frac{1}{m^2}\,\mathbb{E}\left[b^{\perp T}\sum_{i=1}^m s_i s_i^T UU^T \sum_{j=1}^m s_j s_j^T b^\perp\right] \\
&= \frac{1}{m^2}\,\mathbb{E}\left[\sum_{i=1}^m\sum_{j=1}^m b^{\perp T} s_i s_i^T UU^T s_j s_j^T b^\perp\right] \\
&= \frac{1}{m^2}\sum_{1\le i=j\le m} b^{\perp T}\,\mathbb{E}[s_i s_i^T UU^T s_j s_j^T] b^\perp \\
&\quad + \frac{1}{m^2}\sum_{i\neq j,\,1\le i,j\le m} b^{\perp T}\,\mathbb{E}[s_i s_i^T] UU^T\,\mathbb{E}[s_j s_j^T] b^\perp \\
&= \frac{1}{m}\, b^{\perp T}\,\mathbb{E}[s_1 s_1^T UU^T s_1 s_1^T] b^\perp \\
&\quad + \frac{1}{m^2}\sum_{i\neq j,\,1\le i,j\le m} b^{\perp T} I_n UU^T I_n b^\perp \\
&= \frac{1}{m}\, b^{\perp T}\,\mathbb{E}[s_1 s_1^T UU^T s_1 s_1^T] b^\perp \\
&= \frac{1}{m}\, b^{\perp T} n^2 \frac{1}{n}\sum_{i=1}^n e_i e_i^T UU^T e_i e_i^T b^\perp \\
&= \frac{n}{m}\sum_{i=1}^n {b_i^\perp}^2 \|\tilde{u}_i\|_2^2 \\
&\le \frac{n}{m}\sum_{i=1}^n {b_i^\perp}^2 \max_j \|\tilde{u}_j\|_2^2 = \frac{\mu d}{m} f(x^*)\,.
\end{aligned}$$

Next, for uniform sampling without replacement, the rows s_i and s_j are not independent which can be seen by noting that given s_i, we know that s_j will have its nonzero entry at a different place than s_i. Hence, differently from uniform sampling with replacement, the following term will not be zero:

$$\begin{aligned}
&\frac{1}{m^2}\sum_{i\neq j,\,1\le i,j\le m} b^{\perp T}\,\mathbb{E}[s_i s_i^T UU^T s_j s_j^T] b^\perp = \\
&= \frac{m^2-m}{m^2} b^{\perp T}\,\mathbb{E}[s_1 s_1^T UU^T s_2 s_2^T] b^\perp \\
&= \frac{m-1}{m} b^{\perp T} n^2 \frac{1}{n^2-n}\sum_{i\neq j,1\le i,j\le n} e_i e_i^T UU^T e_j e_j^T b^\perp \\
&= \frac{m-1}{m}\frac{n}{n-1} b^{\perp T}\sum_{i\neq j,1\le i,j\le n} e_i \tilde{u}_i^T \tilde{u}_j e_j^T b^\perp \\
&= \frac{m-1}{m}\frac{n}{n-1} b^{\perp T}(UU^T - \mathrm{diag}(\|\tilde{u}_i\|_2^2) b^\perp \\
&= -\frac{m-1}{m}\frac{n}{n-1}\sum_{i=1}^n {b_i^\perp}^2 \|\tilde{u}_i\|_2^2\,.
\end{aligned}$$

It follows that for uniform sampling without replacement, we obtain

$$\begin{aligned}
\mathbb{E}[\|z\|_2^2] &= \left(\frac{n}{m} - \frac{m-1}{m}\frac{n}{n-1}\right)\sum_{i=1}^n {b_i^\perp}^2 \|\tilde{u}_i\|_2^2 \\
&= \frac{n}{m}\frac{n-m}{n-1}\sum_{i=1}^n {b_i^\perp}^2 \|\tilde{u}_i\|_2^2 \\
&\le \frac{n}{m}\frac{n-m}{n-1} f(x^*)\max_i \|\tilde{u}_i\|_2^2 \\
&= \frac{\mu d}{m}\frac{n-m}{n-1} f(x^*)\,.
\end{aligned}$$

■

Proof of Lemma A.5: We consider leverage score sampling with replacement. The rows s_i, s_j $i \neq j$ are independent because sampling is with replacement. We will assume that all the expectations are conditioned on the event E, which we defined earlier as $(1-\epsilon)I_d \preceq Q \preceq (1+\epsilon)I_d$. For leverage score sampling, the term $\mathbb{E}[\|z\|_2^2]$ is upper bounded as follows:

$$\begin{aligned}
\mathbb{E}[\|z\|_2^2] &= \frac{1}{m^2}\,\mathbb{E}\left[b^{\perp T}\sum_{i=1}^m s_i s_i^T UU^T \sum_{j=1}^m s_j s_j^T b^\perp\right] \\
&= \frac{1}{m^2}\,\mathbb{E}\left[\sum_{i=1}^m\sum_{j=1}^m b^{\perp T} s_i s_i^T UU^T s_j s_j^T b^\perp\right] \\
&= \frac{1}{m^2}\sum_{1\le i=j\le m} b^{\perp T}\,\mathbb{E}[s_i s_i^T UU^T s_j s_j^T] b^\perp \\
&\quad + \frac{1}{m^2}\sum_{i\neq j,\,1\le i,j\le m} b^{\perp T}\,\mathbb{E}[s_i s_i^T] UU^T\,\mathbb{E}[s_j s_j^T] b^\perp \\
&= \frac{1}{m}\, b^{\perp T}\sum_{i=1}^n \frac{\ell_i}{d}\frac{d}{\ell_i} e_i e_i^T UU^T \frac{d}{\ell_i} e_i e_i^T b^\perp \\
&\quad + \frac{m^2-m}{m^2} b^{\perp T} I_n UU^T I_n b^\perp \\
&= \frac{1}{m}\, b^{\perp T}\sum_{i=1}^n \frac{d}{\ell_i}\ell_i e_i e_i^T b^\perp \\
&= \frac{d}{m}\,\|b^\perp\|_2^2 = \frac{d}{m} f(x^*)\,.
\end{aligned}$$

■

Proof of Theorem A.6: We will begin by using the results from Table 5 of [2] for selecting the sketch size:

$$\begin{aligned}
&m = O\Big(\frac{d+\log(n)}{\epsilon^2}\log(d/\delta)\Big) \text{ for rand. Hadamard sketch} \\
&m = O\Big(\frac{\mu d}{\epsilon^2}\log(d/\delta)\Big) \text{ for uniform sampling} \\
&m = O\Big(\frac{d}{\epsilon^2}\log(d/\delta)\Big) \text{ for leverage score sampling} \qquad (121)
\end{aligned}$$

where μ is the row coherence of U as defined before. When the sketch sizes are selected according to the formulas above, the subspace embedding property given by

$$\|U^T S^T SU - I_d\|_2 \le \epsilon \qquad (122)$$

is satisfied with probability at least $1-\delta$. Note that the subspace embedding property can be rewritten as

$$(1-\epsilon)I_d \preceq U^T S^T SU \preceq (1+\epsilon)I_d\,. \qquad (123)$$

This implies the following relation for the inverse matrix $(U^T S^T SU)^{-1}$:

$$\frac{1}{1+\epsilon} I_d \preceq (U^T S^T SU)^{-1} \preceq \frac{1}{1-\epsilon} I_d \,. \tag{124}$$

Observe that $1-\epsilon \leq \frac{1}{1+\epsilon}$ for any $\epsilon > 0$ and that $\frac{1}{1-\epsilon} \leq 1+2\epsilon$ for $0 \leq \epsilon \leq 0.5$. Assuming that $0 \leq \epsilon \leq 0.5$, we have

$$(1-2\epsilon) I_d \preceq (U^T S^T SU)^{-1} \preceq (1+2\epsilon) I_d \,. \tag{125}$$

We can rescale $\epsilon \leftarrow \epsilon/2$ so that

$$(1-\epsilon) I_d \preceq (U^T S^T SU)^{-1} \preceq (1+\epsilon) I_d \tag{126}$$

and the effect of this rescaling will be hidden in the O notation for the sketch size formulas.

We will define the events E_k, $k = 1, \ldots, q$ as $(1-\epsilon) I_d \preceq (U^T S_k^T S_k U)^{-1} \preceq (1+\epsilon) I_d$ and it follows that when the sketch sizes are selected according to (121), we will have

$$P(E_k) \geq 1 - \delta \,, k = 1, \ldots, q \,. \tag{127}$$

Now, we find a simpler expression for the error of the single-sketch estimator:

$$\begin{aligned} A\hat{x} - Ax^* &= A(A^T S^T SA)^{-1} A^T S^T S^T b - Ax^* \\ &= A(A^T S^T SA)^{-1} A^T S^T S^T (b^\perp + Ax^*) - Ax^* \\ &= A(A^T S^T SA)^{-1} A^T S^T S^T b^\perp \\ &= U(U^T S^T SU)^{-1} U^T S^T S b^\perp \\ &= UQz \end{aligned} \tag{128}$$

where we defined $Q := (U^T S^T SU)^{-1}$, and $z := U^T S^T S b^\perp$. The variance term is equal to

$$\mathbb{E}[\|A\hat{x} - Ax^*\|_2^2] = \mathbb{E}[\|UQz\|_2^2] = \mathbb{E}[\|Qz\|_2^2] \,. \tag{129}$$

Conditioned on the event E, we can bound the expectation $\mathbb{E}[\|Qz\|_2^2 | E]$ as follows:

$$\begin{aligned} \mathbb{E}[\|Qz\|_2^2 | E] &= \mathbb{E}[z^T Q^T Q z | E] \\ &\leq \mathbb{E}[(1+\epsilon)^2 \|z\|_2^2 | E] \\ &= (1+\epsilon)^2 \, \mathbb{E}[\|z\|_2^2 | E] \end{aligned} \tag{130}$$

where we have used $Q^T Q \preceq (1+\epsilon)^2 I_d$, which follows from all eigenvalues of Q being less than $(1+\epsilon)$ when conditioned on the event E. Next, from Lemma A.1, we obtain the following bound for the expected error

$$\begin{aligned} &\mathbb{E}[f(\bar{x}) - f(x^*) | E_1, \ldots, E_q] = \\ &= \frac{1}{q} \, \mathbb{E}\left[\|A\hat{x} - Ax^*\|_2^2 | E\right] + \frac{q-1}{q} \| \, \mathbb{E}[A\hat{x} - Ax^* | E] \|_2^2 \\ &\leq \frac{1}{q}(1+\epsilon^2) \, \mathbb{E}[\|z\|_2^2 | E] + \frac{q-1}{q} 4\epsilon \, \mathbb{E}[\|z\|_2^2 | E] \,. \end{aligned} \tag{131}$$

Using Markov's inequality gives us the following probability bound:

$$\begin{aligned} &P(f(\bar{x}) - f(x^*) \geq \gamma | E_1, \ldots, E_q) \\ &\leq \frac{1}{\gamma} \, \mathbb{E}[f(\bar{x}) - f(x^*) | E_1, \ldots, E_q] \\ &\leq \frac{1}{\gamma q} \big((1+\epsilon)^2 + 4\epsilon(q-1) \big) \, \mathbb{E}[\|z\|_2^2 | E] \,. \end{aligned} \tag{132}$$

The unconditioned probability can be computed as

$$\begin{aligned} &P(f(\bar{x}) - f(x^*) \geq \gamma) = \\ &= P(f(\bar{x}) - f(x^*) \geq \gamma | E_1, \ldots, E_q) P(\cap_{k=1}^q E_k) \\ &\quad + P(f(\bar{x}) - f(x^*) \geq \gamma | \cup_{k=1}^q E_k^C) P(\cup_{k=1}^q E_k^C) \\ &\leq \frac{1}{\gamma q} \big((1+\epsilon)^2 + 4\epsilon(q-1) \big) \, \mathbb{E}[\|z\|_2^2 | E] + q\delta \end{aligned} \tag{133}$$

where we have used $P(\cup_{k=1}^q E_k^C) \leq q\delta$ and $P(\cap_{k=1}^q E_k) \leq 1$. We can obtain a bound for the relative error as follows:

$$\begin{aligned} &P\Big(\frac{f(\bar{x})}{f(x^*)} \geq 1 + \frac{\gamma}{f(x^*)} \Big) \\ &\quad \leq \frac{1}{\gamma q} \big((1+\epsilon)^2 + 4\epsilon(q-1) \big) \, \mathbb{E}[\|z\|_2^2 | E] + q\delta \,. \end{aligned} \tag{134}$$

Scaling $\gamma \leftarrow \gamma f(x^*)$ leads to

$$\begin{aligned} &P\Big(\frac{f(\bar{x})}{f(x^*)} \leq 1 + \gamma \Big) \\ &\quad \geq 1 - q\delta - \frac{1}{\gamma f(x^*) q} \big((1+\epsilon)^2 + 4\epsilon(q-1) \big) \, \mathbb{E}[\|z\|_2^2 | E] \,. \end{aligned} \tag{135}$$

We now plug the bounds for $\mathbb{E}[\|z\|_2^2 | E]$ from Lemma A.3, A.4, A.5 in the above bound and obtain the following lower bounds for $P\Big(\frac{f(\bar{x})}{f(x^*)} \leq 1 + \gamma \Big)$:

- **Randomized Hadamard sketch:**

$$1 - q\delta - \frac{d}{q\gamma m} \big((1+\epsilon)^2 + 4\epsilon(q-1) \big) \,. \tag{136}$$

- **Uniform sampling with replacement:**

$$1 - q\delta - \frac{\mu d}{q\gamma m} \big((1+\epsilon)^2 + 4\epsilon(q-1) \big) \,. \tag{137}$$

- **Uniform sampling without replacement:**

$$1 - q\delta - \frac{\mu d}{q\gamma m} \frac{n-m}{n-1} \big((1+\epsilon)^2 + 4\epsilon(q-1) \big) . \tag{138}$$

- **Leverage score sampling:**

$$1 - q\delta - \frac{d}{q\gamma m} \big((1+\epsilon)^2 + 4\epsilon(q-1) \big) \,. \tag{139}$$

■

APPENDIX C
ADDITIONAL NUMERICAL RESULTS

In this section, we present additional experimental results.

A. Scalability of the Serverless Implementation

Figure 11 shows the cost against time when we solve the problem given in (37) for large scale data on AWS Lambda using the distributed Newton sketch algorithm. The setting in this experiment is such that each worker node has access to a different subset of data, and there is no additional sketching applied. The dataset used is randomly generated and the goal here is to demonstrate the scalability of the algorithm and the serverless implementation. The size of the data matrix A is 44 GB.

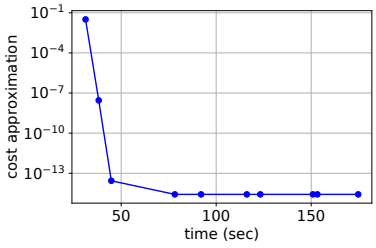
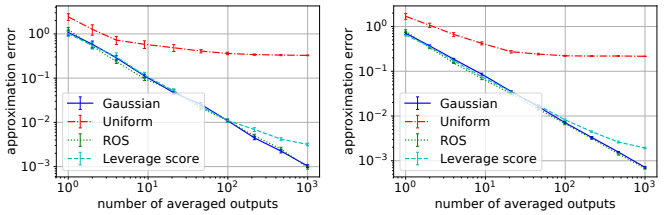


Fig. 11. Cost approximation vs time when we solve the problem given in (37) for a large scale randomly generated dataset (44 GB sized) on AWS Lambda. Circles correspond to times that the iterates x_t are computed. Problem parameters are as follows: $n = 200000$, $d = 30000$, $\lambda_1 = 1$, $m = 2000$, $q = 100$, $\lambda = 10$.



a) echocardiogram b) oocytes merluccius nucleus

Fig. 12. Approximation error against the number of averaged outputs in log-log scale for various sketching methods on two UCI datasets. All of the curves have been averaged over 25 independent trials and the vertical error bars show the standard error. The parameters are as follows. Plot a: $n = 131$, $d = 10$, $m = 20$. Plot b: $n = 1022$, $d = 41$, $m = 100$.

In the serverless computing implementation, we reuse the serverless functions during the course of the algorithm, meaning that the same $q = 100$ functions are used for every iteration. We note that every iteration requires two rounds of communication with the master node. The first round is for the communication of the local gradients, and the second round is for the approximate update directions. The master node, also a serverless function, is also reused across iterations. Figure 11 illustrates that each iteration takes a different amount of time and iteration times can be as short as 5 seconds. The reason for some iterations taking longer times is what is referred to as the straggler problem, which is a phenomenon commonly encountered in distributed computing. More precisely, the iteration time is determined by the slowest of the $q = 100$ nodes and nodes often slow down for a variety of reasons causing stragglers. A possible solution to the issue of straggling nodes is to use error correcting codes to insert redundancy to computation and hence to avoid waiting for the outputs of all of the worker nodes [43], [44]. We identify that implementing straggler mitigation for solving large scale problems via approximate second order optimization methods such as distributed Newton sketch is a promising direction.

B. Experiments on UCI Datasets

In the case of large datasets and limited computing resources of worker nodes such as memory and lifetime, most of the standard sketches are computationally too expensive as discussed in the main body of the paper. This is the reason why we limited the scope of the large scale experiments to uniform sampling, SJLT, and hybrid sketch. In this section we present some additional experimental results on smaller datasets to empirically verify the theoretical results of the paper.

We present results on two UCI datasets in Figure 12 comparing the performances of the sketches we discussed in the paper.

Figure 12 shows that Gaussian and ROS sketches lead to unbiased estimators in the experiments because the corresponding curves appear linear in the log-log scale plots. These experiment results suggest that the upper bound that we have found for the bias of the ROS sketch may not be tight. We see that the estimates for uniform sampling and leverage score sampling are biased. The observation that the Gaussian sketch estimator is unbiased in the experiments is perfectly consistent with our theoretical findings. Furthermore, we observe that the approximation error is the highest in uniform sampling, which is also in agreement with the theoretical upper bounds that we have presented.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation (NSF) under grants ECCS- 2037304, DMS-2134248, NSF CAREER Award CCF-2236829, the U.S. Army Research Office Early Career Award W911NF-21-1-0242, Stanford Precourt Institute, and the ACCESS – AI Chip Center for Emerging Smart Systems, sponsored by InnoHK funding, Hong Kong SAR.

REFERENCES

[1] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, "Faster least squares approximation," *Numerische mathematik*, vol. 117, no. 2, pp. 219–249, 2011.

[2] S. Wang, A. Gittens, and M. W. Mahoney, "Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 8039–8088, Jan. 2017.

[3] M. Pilanci and M. J. Wainwright, "Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1842–1879, 2016.

[4] J. Lacotte and M. Pilanci, "Faster least squares optimization," *arXiv preprint, arXiv:1911.02675*, 2019.

[5] M. W. Mahoney, "Randomized algorithms for matrices and data," *Foundations and Trends® in Machine Learning*, vol. 3, no. 2, pp. 123–224, 2011.

[6] M. Pilanci and M. J. Wainwright, "Randomized sketches of convex programs with sharp guarantees," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 5096–5115, 2015.

[7] ——, "Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence," *SIAM Journal on Optimization*, vol. 27, no. 1, pp. 205–245, 2017.

[8] C.-C. Wang, K. L. Tan, C.-T. Chen, Y.-H. Lin, S. S. Keerthi, D. Mahajan, S. Sundararajan, and C.-J. Lin, "Distributed newton methods for deep neural networks," *Neural Comput.*, vol. 30, no. 6, p. 1673–1724, Jun. 2018. [Online]. Available: https://doi.org/10.1162/neco_a_01088

[9] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[10] F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild! a lock-free approach to parallelizing stochastic gradient descent," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11. Red Hook, NY, USA: Curran Associates Inc., 2011, p. 693–701.

[11] E. Jonas, Q. Pu, S. Venkataraman, I. Stoica, and B. Recht, "Occupy the cloud: Distributed computing for the 99%," in *Proceedings of the 2017 Symposium on Cloud Computing*, ser. SoCC '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 445–451. [Online]. Available: https://doi.org/10.1145/3127479.3128601

[12] J. Carreira, P. Fonseca, A. Tumanov, A. Zhang, , and R. Katz, "A case for serverless machine learning," *Workshop on Systems for ML and Open Source Software at NeurIPS 2018*, 2018.

[13] V. Gupta, S. Phade, T. Courtade, and K. Ramchandran, "Utility-based resource allocation and pricing for serverless computing," 2020. [Online]. Available: https://arxiv.org/abs/2008.07793

[14] S. Zhou, J. Lafferty, and L. Wasserman, "Compressed and privacy-sensitive sparse regression," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 846–866, Feb 2009.

[15] M. Showkatbakhsh, C. Karakus, and S. Diggavi, "Privacy-utility trade-off of linear regression under random projections and additive noise," in *2018 IEEE International Symposium on Information Theory (ISIT)*, June 2018, pp. 186–190.

[16] S. Zhou, J. Lafferty, and L. Wasserman, "Compressed regression," in *Neural Information Processing Systems*, December 2007.

[17] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "The johnson-lindenstrauss transform itself preserves differential privacy," in *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, 2012, pp. 410–419.

[18] S. S. Vempala, *The random projection method*. American Mathematical Soc., 2005, vol. 65.

[19] D. P. Woodruff *et al.*, "Sketching as a tool for numerical linear algebra," *Foundations and Trends® in Theoretical Computer Science*, vol. 10, no. 1–2, pp. 1–157, 2014.

[20] P. Drineas and M. W. Mahoney, "RandNLA: randomized numerical linear algebra," *Communications of the ACM*, vol. 59, no. 6, pp. 80–90, 2016.

[21] H. Avron, P. Maymounkov, and S. Toledo, "Blendenpik: Supercharging lapack's least-squares solver," *SIAM Journal on Scientific Computing*, vol. 32, no. 3, pp. 1217–1236, 2010.

[22] V. Rokhlin, A. Szlam, and M. Tygert, "A randomized algorithm for principal component analysis," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1100–1124, 2009.

[23] S. Wang, F. Roosta, P. Xu, and M. W. Mahoney, "Giant: Globally improved approximate newton method for distributed optimization," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 2332–2342.

[24] T. Sarlos, "Improved approximation algorithms for large matrices via random projections," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, 2006, pp. 143–152.

[25] N. Ailon and B. Chazelle, "Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform," in *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM, 2006, pp. 557–563.

[26] M. W. Mahoney, "Lecture notes on randomized linear algebra," *CoRR*, vol. abs/1608.04481, 2016. [Online]. Available: http://arxiv.org/abs/1608.04481

[27] P. Drineas, M. Magdon-Ismail, M. Mahoney, and D. Woodruff, "Fast approximation of matrix coherence and statistical leverage," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3475–3506, 2012.

[28] J. Nelson and H. L. Nguyên, "Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings," in *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*. IEEE, 2013, pp. 117–126.

[29] G. Letac and H. Massam, "All invariant moments of the wishart distribution," *Scandinavian Journal of Statistics*, vol. 31, no. 2, pp. 295–318, 2004.

[30] D. C. Ahfock, W. J. Astle, and S. Richardson, "Statistical properties of sketching algorithms," *Biometrika*, vol. 108, no. 2, pp. 283–297, 07 2020. [Online]. Available: https://doi.org/10.1093/biomet/asaa062

[31] S. Sridhar, M. Pilanci, and A. Özgür, "Lower bounds and a near-optimal shrinkage estimator for least squares using random projections," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 3, pp. 660–668, 2020.

[32] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

[33] N. E. Karoui, "Concentration of measure and spectra of random matrices: Applications to correlation matrices, elliptical distributions and beyond," *The Annals of Applied Probability*, vol. 19, no. 6, pp. 2362–2405, 2009.

[34] M. Derezinski, B. Bartan, M. Pilanci, and M. W. Mahoney, "Debiasing distributed second order optimization with surrogate sketching and scaled regularization," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 6684–6695. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/4a46fbfca3f1465a27b210f4bdfe6ab3-Paper.pdf

[35] S. Liu and E. Dobriban, "Ridge regression: Structure, cross-validation, and sketching," in *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[36] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques*, ser. EUROCRYPT'06. Berlin, Heidelberg: Springer-Verlag, 2006, p. 486–503. [Online]. Available: https://doi.org/10.1007/11761679_29

[37] O. Sheffet, "Private approximations of the 2nd-moment matrix using existing techniques in linear regression," *CoRR*, vol. abs/1507.00056, 2015. [Online]. Available: http://arxiv.org/abs/1507.00056

[38] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, ser. ICDCN '15. New York, NY, USA: Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2684464.2684480

[39] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[40] B. of Transportation Statistics, "Airline on-time statistics and delay causes, http://stat-computing.org/dataexpo/2009/," 2018.

[41] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2921–2926.

[42] A. A. Borovkov, *Mathematical statistics*. Australia: Gordon and Breach Science Publishers, 1998.

[43] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, March 2018.

[44] B. Bartan and M. Pilanci, "Straggler resilient serverless computing based on polar codes," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 276–283.

Burak Bartan Burak Bartan is a PhD student at Electrical Engineering Department at Stanford University. He received an MS degree in Electrical Engineering from Stanford University in 2018 and a BS degree in Electrical and Electronics Engineering from Bilkent University in 2016. His academic interests include machine learning, optimization, signal processing, distributed computing, and randomized algorithms.

Mert Pilanci Mert Pilanci is an assistant professor of Electrical Engineering at Stanford University. He received his Ph.D. in Electrical Engineering and Computer Science from UC Berkeley in 2016. Prior to joining Stanford, he was an assistant professor of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor from 2017-2018. He was a Math+X postdoctoral fellow at Stanford University in 2017. His research interests are in machine learning, convex optimization, neural networks and information theory.