Iterative Sketching for Secure Coded Regression

Neophytos Charalambides, Hessam Mahdavifar, Mert Pilanci, and Alfred O. Hero III

Abstract

In this work, we propose methods for speeding up linear regression distributively, while ensuring security. We leverage randomized sketching techniques, and improve straggler resilience in asynchronous systems. Specifically, we apply a random orthonormal matrix and then subsample *blocks*, to simultaneously secure the information and reduce the dimension of the regression problem. In our setup, the transformation corresponds to an encoded encryption in an *approximate gradient coding scheme*, and the subsampling corresponds to the responses of the non-straggling workers; in a centralized coded computing network. This results in a distributive *iterative sketching* approach for an ℓ_2-subspace embedding, *i.e.* a new sketch is considered at each iteration. We also focus on the special case of the *Subsampled Randomized Hadamard Transform*, which we generalize to block sampling; and discuss how it can be modified in order to secure the data.

Index Terms

Coded computing, gradient coding, subspace embedding, linear regression, secure computing, iterative sketching.

I. INTRODUCTION

Random projections[1] are a classical way of performing dimensionality reduction, and are widely used in algorithmic and learning contexts [?], [?], [?], [?]. Distributed computations in the presence of *stragglers* have gained a lot of attention in the information theory community. Coding-theoretic approaches have been adopted for this [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], and fall under the framework of *coded computing* (CC). Data security is also an increasingly important issue in CC [?]. In this work, we propose methods to securely speed up linear regression by simultaneously leveraging random projections and sketching, and distributed computations. Our results are presented in terms of the system model proposed in [?], though they extend to any centralized distributed model, *i.e.* distributed systems with a central server which updates and communicates the parameters to the computational worker nodes. Furthermore, the desiderata of our approaches are to:

 (I)   produce straggler-resilient accurate approximations,
 (II) secure the information,
 (III) carry out the computations efficiently and distributively.

We focus on sketching for *steepest descent* (SD) in the context of solving overdetermined linear systems. Part of our theoretical results are given for the sketch-and-solve paradigm [?], which can be utilized by a single server; who can also store a compressed version of the data. The application through CC results in iterative sketching methods for distributively solving overdetermined linear systems. We propose applying a random orthonormal projection to the linear system before distributing the data, and then performing SD distributively on the transformed system through *approximate gradient coding*. Under the straggler scenario and the assumptions we make, this results in a *mini-batch stochastic steepest descent* (SSD) procedure of the original system. A special case of such a projection is the *Subsampled Randomized Hadamard Transform* (SRHT) [?], [?], [?], which utilizes Kronecker products of the Hadamard transform; and relates to the *fast Johnson-Lindenstrauss transform* [?], [?].

The benefit of applying an orthonormal matrix transformation is that we rotate and/or reflect the data's orthonormal basis, which *cannot* be reversed without knowledge of the transformation. This is leveraged to give security guarantees, while simultaneously ensuring that we recover well-approximated gradients, and an approximate solution of the linear system. Such sketching matrices are also referred to as *randomized orthonormal systems* [?]. We also discuss how one can use recursive Kronecker products of an orthonormal matrix of dimension greater than 2 in place of the Hadamard transform, to obtain the a more efficiency encoding and encryption than through a random and unstructured orthonormal matrix.

In the CC paradigm, the workers are assumed to be *homogeneous* with the same expected response time. In the proposed methods, we stop receiving computations once a fixed fraction of the workers respond; producing a different induced sketch at each iteration. A predominant task which has been studied in the CC framework is the gradient computation of differentiable and additively separable objective functions [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?]. These schemes are collectively called *gradient coding* (GC). We note that iterative sketching has proven to be a powerful tool for second-order methods [?], [?], though it has not been explored in first-order methods. Since we indirectly consider a different underlying modified problem at each iteration, the methods we propose are *approximate* GC schemes (GCSs). Related approaches have been proposed in [?], [?], [?], [?], [?], [?], [?], [?], [?], [?]. Two important benefits of our approaches are that we do not require a decoding step, nor an encoding step by the workers; at each iteration, avoiding prevalent bottlenecks.

[1] By '*projections*' we refer to random matrices, not idempotent matrices.

An advantage of using an updated sketch at each iteration, is that we do not have a bias towards the samples that would be selected/returned when the sketching takes place, which occurs in the sketch-and-solve approach. Specifically, we do not solve a modified problem which only accounts for a reduced dimension; determined at the beginning of the iterative process. Instead, we consider a different reduced system at each iteration. This is also justified numerically.

Another benefit of our approach, is that random projections secure the information from potential eavesdroppers, honest but curious; and colluding workers. We show information-theoretic security for the case where a random orthonormal projection is utilized in our sketching algorithm. Furthermore, the security of the SRHT, which is a crucial aspect, has not been extensively studied. Unfortunately, the SRHT is inherently insecure, which we show. We propose a modified projection which guarantees computational security of the SRHT.

There are related works to what we study. The work of [?] focuses on parameter averaging for variance reduction, but only mentions a security guarantee for the Gaussian sketch, derived in [?]. Another line of work [?], [?], focuses on introducing redundancy through equiangular tight frames (ETFs), partitioning the system into smaller linear systems, and then averaging the solutions of a fraction of them. A drawback of using ETFs, is that most of them are over $\mathbb{C}$. The authors of [?] study privacy of random projections, though make the assumption that the projections meet the 'ε-MI-DP constraint'. Recently, the authors of [?] considered CC privacy guarantees through the lens of differential privacy, with a focus on matrix multiplication. Lastly, a secure GCS is studied in [?], though it does not utilize sketching. We also clarify that even though we guarantee cryptographic security, our methods may still be vulnerable to various privacy attacks, *e.g.* membership inference attacks [?] and model inversion attacks [?]. This is another interesting line of work, though is not a focus of our approach.

The paper is organized as follows. In **??** we review the framework and background for coded linear regression, the notions of security we will be working with, the ℓ_2-subspace embedding property, and list the main properties we seek to satisfy through our constructions; in order to meet the aforementioned desiderata. In **??** we present the proposed iterative sketching algorithm, and in **??** the special case where the projection is the randomized Hadamard transform; which we refer to as the "*block-SRHT*". The subspace embedding results for the general algorithm and the block-SRHT are presented in the respective sections. We consider the case where the central server may adaptively change the step-size of its SD procedure in **??**, which can be viewed as an *adaptive GCS*. In **??** we present the security guarantees of our algorithm and the modified version of the block-SRHT. Finally, we present numerical experiments in **??**; and concluding remarks in **??**.

II. Coded Linear Regression

A. *Least Squares Approximation and Steepest Descent*

In linear least squares approximation [?], it is desired to approximate the solution

$$\mathbf{x}_{ls}^{\star} = \arg \min_{\mathbf{x} \in \mathbb{R}^d} \left\{ L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \right\} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N \times d}$ and $\mathbf{b} \in \mathbb{R}^N$. This corresponds to the regression coefficients $\mathbf{x}$ of the model $\mathbf{b} = \mathbf{A}\mathbf{x} + \vec{\varepsilon}$, which is determined by the dataset $\mathcal{D} = \{(\mathbf{a}_i, b_i)\}_{i=1}^N \subsetneq \mathbb{R}^d \times \mathbb{R}$ of N samples, where $(\mathbf{a}_i, b_i)$ represent the features and label of the ith sample, *i.e.* $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_N \end{bmatrix}^T$ and $\mathbf{b} = \begin{bmatrix} b_1 & \cdots & b_N \end{bmatrix}^T$.

To simplify our presentation, we first define some notational conventions. Row vectors of a matrix $\mathbf{M}$ are denoted by $\mathbf{M}_{(i)}$, and column vectors by $\mathbf{M}^{(j)}$. Our embedding results are presented in terms of an arbitrary partition $\mathbb{N}_N = \bigsqcup_{\iota=1}^K \mathcal{K}_\iota$, for $\mathbb{N}_N := \{1, \cdots, N\}$ the index set of $\mathbf{M}$'s rows. Each $\mathcal{K}_\iota$ corresponds to a *block* of $\mathbf{M}$. The notation $\mathbf{M}_{(\mathcal{K}_\iota)}$ denotes the submatrix of $\mathbf{M}$ comprised of the rows indexed by $\mathcal{K}_\iota$. That is: $\mathbf{M}_{(\mathcal{K}_\iota)} = \mathbf{I}_{(\mathcal{K}_\iota)}\mathbf{M}$, for $\mathbf{I}_{(\mathcal{K}_\iota)}$ the corresponding submatrix of $\mathbf{I}_N$ of size $|\mathcal{K}_\iota| \times N$. We call $\mathbf{M}_{(\mathcal{K}_\iota)}$ the 'ι^{th} *block of* $\mathbf{M}$'. We abbreviate $(1 - \epsilon) \cdot \|\vec{b}\| \leqslant \|\vec{a}\| \leqslant (1 + \epsilon) \cdot \|\vec{b}\|$, to $\|\vec{a}\| \leqslant_\epsilon \|\vec{b}\|$. Lastly, '$\leftarrow$' denotes a numerical assignment of a varying quantity, and '$\overset{U}{\leftarrow}$' a realization of a random variable through uniform sampling.

By $\mathbf{\Pi}$ we denote the random orthonormal matrix we apply to the data matrix $\mathbf{A}$; which is drawn uniformly at random from a finite subset $\tilde{O}_{\mathbf{A}}$ of the set orthonormal matrices $O_N(\mathbb{R})$, *i.e.* $\mathbf{\Pi} \overset{U}{\leftarrow} \tilde{O}_{\mathbf{A}} \subsetneq O_N(\mathbb{R})$. By $\hat{\mathbf{\Pi}}$ and $\tilde{\mathbf{\Pi}}$ we denote the special cases where $\mathbf{\Pi}$ is a orthonormal matrix used for the *block-SRHT* and *garbled block-SRHT* respectively.

We address the overdetermined case where $N \gg d$. Existing exact methods find a solution vector $\mathbf{x}_{ls}^\star$ in $\mathcal{O}(Nd^2)$ time, where $\mathbf{x}_{ls}^\star = \mathbf{A}^\dagger \mathbf{b}$. A common way to approximate $\mathbf{x}_{ls}^\star$ is through SD, which iteratively updates the gradient

$$g_{ls}^{[t]} := \nabla_{\mathbf{x}} L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]}) = 2\mathbf{A}^T(\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b}) \tag{2}$$

followed by updating the parameter vector

$$\mathbf{x}^{[t+1]} \leftarrow \mathbf{x}^{[t]} - \xi_t \cdot g_{ls}^{[t]} . \tag{3}$$

In our setting, the step-size $\xi_t > 0$ is determined by the central server. The script $[t]$ indexes the iteration $t = 0, 1, 2, \ldots$ which we drop when clear from the context. In **??**, we derive the optimal step-size $\xi_t^\star$ for **(??)** and the modified problems we consider, given the updated gradients and parameters of iteration t.

B. The Straggler Problem and Gradient Coding

Gradient coding is deployed in centralized computation networks, *i.e.* a central server communicates $\mathbf{x}^{[t]}$ to m workers; who perform computations and then communicate back their results. The central server distributes the dataset $\mathcal{D}$ among the m workers, to facilitate the solution of optimization problems with additively separable and differentiable objective functions. For linear regression (**??**), the data is partitioned as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1^T & \cdots & \mathbf{A}_K^T \end{bmatrix}^T \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1^T & \cdots & \mathbf{b}_K^T \end{bmatrix}^T \tag{4}$$

where $\mathbf{A}_i \in \mathbb{R}^{\tau \times d}$ and $\mathbf{b}_i \in \mathbb{R}^\tau$ for all i, and $\tau = N/K$. For ease of exposition, we assume that $K|N$. Then we have $L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}) = \sum_{i=1}^K L_{ls}(\mathbf{A}_i, \mathbf{b}_i; \mathbf{x})$. A regularizer $\mu R(\mathbf{x})$ can also be added to $L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x})$ if desired.

In GC [**?**], the workers encode their computed *partial gradients* $g_i \coloneqq \nabla_{\mathbf{x}} L_{ls}(\mathbf{A}_i, \mathbf{b}_i; \mathbf{x})$; which are then communicated to the central server. Once a certain fraction of encodings is received, the central server applies a decoding step to recover the gradient $g = \nabla_{\mathbf{x}} L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}) = \sum_{i=1}^K g_i$. This can be computationally prohibitive, and takes place at every iteration. To the best of our knowledge, the lowest decoding complexity is $\mathcal{O}\left((s+1) \cdot \lceil \frac{m}{s+1} \rceil\right)$; where s is the number of stragglers [**?**].

In our approach we trade time; by not requiring encoding nor decoding steps at each iteration, with accuracy of approximating $\mathbf{x}_{ls}^\star$. Unlike conventional GCSs, in this paper the workers carry out the computation on the encoded data. The resulting gradient, is that of the modified least squares problem

$$\hat{\mathbf{x}}_{ls} = \arg \min_{\mathbf{x} \in \mathbb{R}^d} \left\{ L_{\mathbf{S}^{[t]}}(\mathbf{A}, \mathbf{b}; \mathbf{x}) \coloneqq \|\mathbf{S}^{[t]}(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2^2 \right\} \tag{5}$$

for $\mathbf{S}^{[t]} \in \mathbb{R}^{r \times N}$ a sketching matrix, with $r \ll N$ and $r > d$. This is the core idea behind our approximation, where we incorporate iterative sketching with orthonormal matrices and random sampling; and generalizations of the SRHT for $\mathbf{S}^{[t]}$, for our approximate GCSs. The sketching approach we take is to first apply a random projection, which also provides security against the workers and eavesdroppers, and then sample computations carried out on the blocks of the transformed data uniformly at random; which corresponds to the responses of the homogeneous non-stragglers.

For q the total number of responsive workers, we can mitigate up to $s = m - q$ stragglers. Specifically, the number of responsive workers $m - s$ in the CC model, corresponds to the number of sampling trials q of our sketching algorithm, *i.e.* $q = m - s$. At iteration t, a SD update of the modified least squares problem (**??**) is obtained distributively. Furthermore, we assume that the data is partitioned into as many blocks as there are workers, *i.e.* $K = m$. The stragglers are assumed to be uniformly random and may differ at each iteration. Thus, there is a different sketching matrix $\mathbf{S}^{[t]}$ at each epoch.

In conventional GCSs the objective is to construct an encoding matrix $\mathbf{G} \in \mathbb{R}^{m \times K}$ (can have $m \neq K$) and decoding vectors $\mathbf{a}_{\mathcal{I}} \in \mathbb{R}^{1 \times q}$, such that $\mathbf{a}_{\mathcal{I}} \mathbf{G}_{(\mathcal{I})} = \vec{\mathbf{1}}$ for any set of non-straggling workers $\mathcal{I}$. Furthermore, it is assumed that multiple replications of each encoded block is shared among the workers, such that $m \geqslant q \geqslant K$.[2] From the fact that $\mathbf{a}_{\mathcal{I}} \mathbf{G}_{(\mathcal{I})} = \vec{\mathbf{1}}$ for any $\mathcal{I}$, in *approximate* GC [**?**], the optimal decoding vector for a set $\mathcal{I}$ of size $q = m - s$ is determined by

$$\mathbf{a}_{\mathcal{I}}^\star = \arg \min_{\mathbf{a} \in \mathbb{R}^{1 \times q}} \left\{ \|\mathbf{a}\mathbf{G}_{(\mathcal{I})} - \vec{\mathbf{1}}\|_2^2 \right\} \quad \implies \quad \mathbf{a}_{\mathcal{I}}^\star = \vec{\mathbf{1}} \mathbf{G}_{(\mathcal{I})}^\dagger \,,$$

for $\mathbf{G}_{(\mathcal{I})}^\dagger$ the pseudoinverse of $\mathbf{G}_{(\mathcal{I})}$. The error in the approximated gradient $\mathring{g}^{[t]}$ of an optimal approximate linear regression GCS $(\mathbf{G}, \mathbf{a}_{\mathcal{I}}^\star)$, is then

$$\left\| g^{[t]} - \mathring{g}^{[t]} \right\|_2 \leqslant 2\sqrt{K} \cdot \text{err}(\mathbf{G}_{(\mathcal{I})}) \cdot \|\mathbf{A}\|_2 \cdot \|\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b}\|_2 \,, \tag{6}$$

for $\text{err}(\mathbf{G}_{(\mathcal{I})}) \coloneqq \left\| \mathbf{I}_K - \mathbf{G}_{(\mathcal{I})}^\dagger \mathbf{G}_{(\mathcal{I})} \right\|_2$.

C. Secure Coded Computing Schemes

Modern cryptography is split into two main categories, *information-theoretic security* and *computational security*. The former is also referred to as *Shannon secrecy*, while the latter is also referred to as *asymptotic security*. In this subsection, we give the definitions which will allow us to characterize the security level of our GCSs.

Definition 1. *A **secure CC** scheme, is the pair of encoding and decoding algorithms* (Enc, Dec) *of the CC scheme, such that* Enc($\mathbf{A}$) *also guarantees a level of security of $\mathbf{A}$, and* Dec *recovers the hidden information; i.e.* Dec(Enc($\mathbf{A}$)) = $\mathbf{A}$.

In our work, Enc corresponds to a linear transformation through a randomly selected orthonormal matrix $\mathbf{\Pi}$. The orthogonal group in dimension N, is denoted by $O_N(\mathbb{R})$. By encryption, we refer to this linear transformation, which is utilized in our GCS. Furthermore, we do not require a decryption step by the central server, as it computes an unbiased estimate of the gradient at the end of each iteration. Also, since $\mathbf{\Pi}^T = \mathbf{\Pi}^{-1}$, it follows that $\mathbf{\Pi}^T$ meets the requirement of Dec, so in the following definition

---

[2]As we mention in **??**, this can be done in order to mimic sampling *with replacement* through the CC network. The reason we require $q \geqslant K$, is to define $\mathbf{a}_{\mathcal{I}}^\star = \vec{\mathbf{1}} \mathbf{G}_{(\mathcal{I})}^\dagger$. This idea has been extensively studied in [**?**].

we refer to the encoding-decoding pair by only referencing Enc. Furthermore, Enc depends on a secret key k which is randomly generated. In our case, this is simply $\boldsymbol{\Pi}$.

Definition 2 (Ch.2 [?]). *An encryption scheme* Enc *with message, ciphertext and key spaces* $\mathcal{M}$, $\mathcal{C}$ *and* $\mathcal{K}$ *respectively is* **Shannon secret** *w.r.t. a probability distribution D over $\mathcal{M}$, if for all $\bar{m} \in \mathcal{M}$ and all $\bar{c} \in \mathcal{C}$:*

$$\Pr_{\substack{m \sim D \\ k \xleftarrow{U} \mathcal{K}}} [m = \bar{m} \mid \mathsf{Enc}_k(m) = \bar{c}] = \Pr_{m \sim D} [m = \bar{m}] \ . \tag{7}$$

An equivalent condition is **perfect secrecy**, *which states that for all $m_0, m_1 \in \mathcal{M}$:*

$$\Pr_{k \xleftarrow{U} \mathcal{K}} [\mathsf{Enc}_k(m_0) = \bar{c}] = \Pr_{k \xleftarrow{U} \mathcal{K}} [\mathsf{Enc}_k(m_1) = \bar{c}] \ . \tag{8}$$

Definition 3 (Ch.3 [?]). *An encryption scheme is* **computationally secure** *if any probabilistic polynomial-time adversary succeeds in breaking it, with at most negligible probability. By negligible, we mean it is asymptotically smaller than any inverse polynomial function.*

D. The ℓ_2-subspace embedding Property

For the analysis of the sketching matrices $\mathbf{S}_{\boldsymbol{\Pi}}$ we propose in Algorithm **??**, we consider any orthonormal basis $\mathbf{U} \in \mathbb{R}^{N \times d}$ of the column-space of $\mathbf{A}$, *i.e.* $\mathrm{im}(\mathbf{A}) = \mathrm{im}(\mathbf{U})$. The subscript of $\mathbf{S}_{\boldsymbol{\Pi}}$, indicates the dependence of the sketching matrix on $\boldsymbol{\Pi}$.

Recall that the ℓ_2-*subspace embedding* (ℓ_2-s.e.) property [?], [?] states that any $\mathbf{y} \in \mathrm{im}(\mathbf{U})$ satisfies:

$$\|\mathbf{S}_{\boldsymbol{\Pi}} \mathbf{y}\|_2 \leqslant_\epsilon \|\mathbf{y}\|_2 \iff \|\mathbf{I}_d - (\mathbf{S}_{\boldsymbol{\Pi}} \mathbf{U})^T (\mathbf{S}_{\boldsymbol{\Pi}} \mathbf{U})\|_2 \leqslant \epsilon \tag{9}$$

for $\epsilon > 0$. In turn, this characterizes the approximation's error of the solution $\hat{\mathbf{x}}_{ls}$ of (**??**) for $\mathbf{S} \leftarrow \mathbf{S}_{\boldsymbol{\Pi}}$, as

$$\|\mathbf{A}\hat{\mathbf{x}}_{ls} - \mathbf{b}\|_2 \leqslant \frac{1 + \epsilon}{1 - \epsilon} \|\mathbf{A}\mathbf{x}_{ls}^\star - \mathbf{b}\|_2 \leqslant (1 + \mathcal{O}(\epsilon)) \|\mathbf{A}\mathbf{x}_{ls}^\star - \mathbf{b}\|_2$$

with high probability, and $\|\mathbf{A}(\mathbf{x}_{ls}^\star - \hat{\mathbf{x}}_{ls})\|_2 \leqslant \epsilon \|(\mathbf{I}_N - \mathbf{U}\mathbf{U}^T)\mathbf{b}\|_2$.

E. Properties of our Approach

A key property in the construction of our sketching matrices, is to sample *blocks* (*i.e.* submatrices) of a transformation of the data matrix, which permits us to then perform the computations in parallel. The additional properties we seek to satisfy with our GCSs through block sampling are the following:

(a) the underlying sketching matrix satisfies the ℓ_2-s.e. property,
(b) the block leverage scores are flattened through the random projection $\boldsymbol{\Pi}$,
(c) the projection is over $\mathbb{R}$,
(d) the central server computes an unbiased gradient estimate at each iteration,
(e) do not require encoding/decoding at each iteration,
(f) guarantee security of the information from the workers and potential eavesdroppers,
(g) $\boldsymbol{\Pi}$ can be applied efficiently, *i.e.* in $\mathcal{O}(Nd \log N)$ operations.

The seven properties listed above, are grouped together with respect to the desiderata mentioned in **??**. Specifically, desideratum (I) encompasses properties (a), (b), (c), (d), desideratum (II) corresponds to (f), and (III) encompasses (b), (c), (e), (g).

Property (a) is motivated by the sketch-and-solve approach, though through the iterative process, in practice we benefit by having fresh sketches. Leverage scores define the key structural non-uniformity that must be dealt with in developing fast randomized matrix algorithms; and are formally defined in **??**. If property (b) is met, we can then sample uniformly at random in order to guarantee (a). We require $\boldsymbol{\Pi}$ to be over $\mathbb{R}$, as if it were over $\mathbb{C}$, the communication cost from the central server to the workers; and the necessary storage space at the workers would double. Additionally, performing computations over $\mathbb{C}$ would result in further round-off errors and numerical instability. Properties (d) and (e) are met by requiring $\boldsymbol{\Pi}$ to be an orthonormal matrix. By allowing the projection to be random; we can secure the data, *i.e.* satisfy (f). Furthermore, the action of applying an orthonormal projection for our encryption; is reversed through the computation of the partial gradients, hence no decryption step is required.

By considering a larger ensemble of orthonormal projections to select from, we can give stronger security guarantees. Specifically, by not restricting the structure of $\boldsymbol{\Pi}$, we can guarantee Shannon secrecy, though this prevents us from satisfying (g). On the other hand, if we let $\boldsymbol{\Pi}$ be structured, we can satisfy (g) at the cost of only guaranteeing computational security.

We point out that even though Gaussian and random Rademacher sketches satisfy satisfy (a), (b), (c) and (f), they do not satisfy (d), (e) nor (g) in our CC setting. Experimentally, we observe that our proposed sketching matrices outperform the Gaussian and random Rademacher sketches, primarily due to the fact that (d) is satisfied. Furthermore, for $\boldsymbol{\Pi} \in O_N(\mathbb{R})$, our distributive procedure results in a SSD approach.

III. BLOCK SUBSAMPLED ORTHONORMAL SKETCHES

Sampling blocks for sketching least squares has not been explored as extensively as sampling rows, though there has been interest in using "block-iterative methods" for solving systems of linear equations [?], [?], [?], [?]. Our interest in sampling blocks, is to invoke results and techniques from *randomized numerical linear algebra* (RandNLA) to CC. Specifically, we apply the transformation before partitioning the data and sharing it between the workers, who will compute the respective partial gradients. Then, the slowest s workers will be disregarded. The proposed sketching matrices are summarised in Algorithm **??**.

Algorithm 1: Subsampled Orthonormal Sketches

Input: $\mathbf{A} \in \mathbb{R}^{N \times d}$, $\mathbf{b} \in \mathbb{R}^N$, $\mathbf{x}^{[0]} \in \mathbb{R}^d$, $\tau = \frac{N}{K}$, $q = \frac{r}{\tau} > \frac{d}{\tau}$
Output: approximate solution $\hat{\mathbf{x}} \in \mathbb{R}^d$ to (**??**)
Randomly Select: $\mathbf{\Pi} \in O_N(\mathbb{R})$, an orthonormal matrix
for $t = 0, 1, 2, \ldots$ **do**
    Initialize: $\mathbf{\Omega} = \mathbf{0}_{q \times K}$
    Select: step-size $\xi_t > 0$
    for $i = 1$ *to* q **do**
        uniformly sample with replacement j_i from $\mathbb{N}_K$
        $\mathbf{\Omega}_{i,j_i} = \sqrt{N/r} = \sqrt{K/q}$
    end
    $\widetilde{\mathbf{\Omega}}_{[t]} \leftarrow \mathbf{\Omega} \otimes \mathbf{I}_\tau$ $\triangleright \mathbf{S}_{\mathbf{\Pi}}^{[t]} = \widetilde{\mathbf{\Omega}}_{[t]} \cdot \mathbf{\Pi}$
    $\widehat{\mathbf{A}}_{[t]} \leftarrow \widetilde{\mathbf{\Omega}}_{[t]} \cdot (\mathbf{\Pi A}) = \mathbf{S}_{\mathbf{\Pi}}^{[t]} \cdot \mathbf{A}$
    $\widehat{\mathbf{b}}_{[t]} \leftarrow \widetilde{\mathbf{\Omega}}_{[t]} \cdot (\mathbf{\Pi b}) = \mathbf{S}_{\mathbf{\Pi}}^{[t]} \cdot \mathbf{b}$
    Update: $\hat{\mathbf{x}}^{[t+1]} \leftarrow \hat{\mathbf{x}}^{[t]} - \xi_t \cdot \nabla_{\mathbf{x}} L_{ls}\left(\widehat{\mathbf{A}}_{[t]}, \widehat{\mathbf{b}}_{[t]}; \hat{\mathbf{x}}^{[t]}\right)$
end

To construct the sketch $\widehat{\mathbf{A}}$, we first transform the orthonormal basis $\mathbf{U}$ by applying $\mathbf{\Pi}$ to $\mathbf{A}$. Then, we subsample q many blocks from $\mathbf{\Pi A}$, to reduce the dimension. Finally, we normalize by $\sqrt{N/r}$ to reduce the variance of the estimator $\widehat{\mathbf{A}}$. Analogous steps are carried out on $\mathbf{\Pi b}$, to construct $\widehat{\mathbf{b}}$.

A. Distributed Steepest Descent and Iterative Sketching

We now discuss the workers' computational tasks of our proposed GCS, when SD is carried out distributively. The encoding corresponds to $\tilde{\mathbf{A}} = \mathbf{G} \cdot \mathbf{A}$ and $\tilde{\mathbf{b}} = \mathbf{G} \cdot \mathbf{b}$ for $\mathbf{G} := \sqrt{N/r} \cdot \mathbf{\Pi}$, which are then partitioned into K encoded block pairs $(\tilde{\mathbf{A}}_i, \tilde{\mathbf{b}}_i)$; similar to (**??**), and are sent to distinct workers. Specifically, $\tilde{\mathbf{A}}_i = \mathbf{I}_{(\mathcal{K}_i)}(\mathbf{GA})$ and $\tilde{\mathbf{b}}_i = \mathbf{I}_{(\mathcal{K}_i)}(\mathbf{Gb})$. This differs from most GCSs, in that the encoding is usually done locally by the workers on the computed results; at each iteration.

If each worker respectively computes $\nabla_{\mathbf{x}} L_{ls}(\tilde{\mathbf{A}}_i, \tilde{\mathbf{b}}_i; \mathbf{x}^{[t]}) = 2\tilde{\mathbf{A}}_i^T(\tilde{\mathbf{A}}_i^T \mathbf{x}^{[t]} - \tilde{\mathbf{b}}_i)$ at iteration t, and the index set of the first q responsive workers is $\mathcal{S}^{[t]}$, the aggregated gradient

$$\hat{g}^{[t]} = 2 \sum_{j \in \mathcal{S}^{[t]}} \tilde{\mathbf{A}}_j^T \left(\tilde{\mathbf{A}}_j \mathbf{x}^{[t]} - \tilde{\mathbf{b}}_j \right) \tag{10}$$

is equal to the gradient of $L_{\mathbf{S}}$ for $\mathbf{S} \leftarrow \mathbf{S}_{\mathbf{\Pi}}^{[t]}$ the induced sketching matrix at that iteration, *i.e.* $\hat{g}^{[t]} = \nabla_{\mathbf{x}} L_{\mathbf{S}_{\mathbf{\Pi}}^{[t]}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$. The sampling matrix $\widetilde{\mathbf{\Omega}}_{[t]}$ and index set $\mathcal{S}^{[t]}$, correspond to the q responsive workers. We illustrate our procedure in Figure **??**.
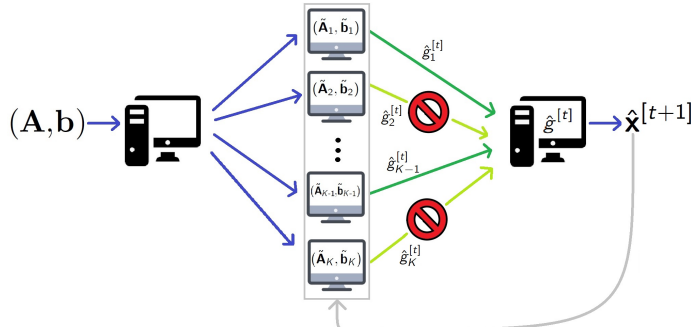

Fig. 1. Illustration of our iterative sketching based GCS, at epoch $t + 1$.

In Algorithm **??**, Theorems **??** and **??**, we assume sampling with replacement. In what we just described, we used one replica of each block, thus $K = m$. To compensate for this, more than one replica of each block could be distributed. This is not a major concern with uniform sampling, as the probability that the ith block would be sampled more than once is $(q-1)/K^2$, which is negligible for large K.

Lemma 1. *At any iteration t, with no replications of the blocks across the network, the resulting sketching matrix $\mathbf{S}_{[t]}$ satisfies* $\mathbb{E}\left[\mathbf{S}_{[t]}^T \mathbf{S}_{[t]}\right] = \mathbb{E}\left[\widetilde{\boldsymbol{\Omega}}_{[t]}^T \widetilde{\boldsymbol{\Omega}}_{[t]}\right] = \mathbf{I}_N$.

It is worth noting that by Lemma **??**, $\mathbf{S}_{[t]}$ in expectation satisfies the ℓ_2-s.e. identity (**??**) with $\epsilon = 0$, as

$$\mathbb{E}\left[\mathbf{U}^T(\mathbf{S}_{[t]}^T \mathbf{S}_{[t]})\mathbf{U}\right] = \mathbf{U}^T \mathbb{E}\left[\mathbf{S}_{[t]}^T \mathbf{S}_{[t]}\right] \mathbf{U} = \mathbf{U}^T \mathbf{U} = \mathbf{I}_d .$$

Theorem 1. *The proposed GCS results in a mini-batch stochastic steepest descent procedure for*

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ L_{\mathbf{G}}(\mathbf{A}, \mathbf{b}; \mathbf{x}) := L_{ls}(\mathbf{GA}, \mathbf{Gb}; \mathbf{x}) \right\} . \tag{11}$$

Moreover $\mathbb{E}\left[\hat{g}^{[t]}\right] = \frac{q}{K} \cdot g_{ls}^{[t]}$.

Lemma 2. *The optimal solution of the modified least squares problem $L_{\mathbf{G}}$, is equal to the optimal solution $\mathbf{x}_{ls}^\star$ of (**??**).*

To prove Theorem **??**, note that $\widetilde{\boldsymbol{\Omega}}_{[t]}$ corresponds to a uniform random selection of q out of K batches for each t; as in SSD, while in our procedure we consider the partial gradients of the q fastest responses. When computing $\nabla_{\mathbf{x}} L_{\mathbf{G}}$, the factor $\boldsymbol{\Pi}$ is annihilated; and the scaling factor $\sqrt{K/q}$ is squared.

Since $\mathbb{E}\left[\hat{g}^{[t]}\right] = \frac{q}{K} g_{ls}^{[t]}$, the estimate $\hat{g}^{[t]}$ is unbiased after an appropriate rescaling; which could be incorporated in the step-size ξ_t. By Theorem **??** and Lemma **??**, it follows that with a diminishing step-size, our updates $\hat{\mathbf{x}}^{[t]}$ converge to $\mathbf{x}_{ls}^\star$ in expectation; at a rate of $\mathcal{O}(1/\sqrt{t} + r/t)$ [**?**], [**?**].

Corollary 1. *Consider the problems (**??**) and (**??**), which are respectively solved through SD and our iterative sketching based GCS. Assume that the two approaches have the same starting point $\mathbf{x}^{[0]}$ and index set $\mathcal{S}^{[t]}$ at each t; and $\hat{\xi}_t = \frac{K}{q}\xi_t$ the step-sizes used for our scheme. Then, in expectation, our approach through Algorithm **??** has the same update at each step t as SD at the corresponding update, i.e $\mathbb{E}\left[\hat{\mathbf{x}}^{[t]}\right] = \mathbf{x}^{[t]}$.*

By Lemma **??** and Corollary **??**, the updated parameter estimates $\hat{\mathbf{x}}^{[0]}, \hat{\mathbf{x}}^{[1]}, \hat{\mathbf{x}}^{[2]}, \ldots$ of Algorithm **??** approach the optimal solution $\mathbf{x}_{ls}^\star$ of (**??**), by solving the modified regression problem (**??**) through SSD. It is also worth noting that the contraction rate of our GC approach, in expectation is equal to that of regular SD. This can be shown through an analogous derivation of [**?**, Theorem 6].

In the next subsection, we present our main ℓ_2-s.e. result.

*B. Subspace Embedding of Algorithm **??***

To give an embedding guarantee for Algorithm **??**, we first show that the block leverage scores of $\boldsymbol{\Pi}\mathbf{A}$ are "flattened", *i.e.* they are all approximately equal. This is precisely what allows us to sample blocks for the construction of $\mathbf{S}_{\boldsymbol{\Pi}}$; and in the distributed approach the computations, *uniformly* at random. Recall that the *leverage scores* of $\tilde{\mathbf{U}} := \boldsymbol{\Pi}\mathbf{U}$ are $\ell_i := \|\tilde{\mathbf{U}}_{(i)}\|_2^2$ for $i \in \mathbb{N}_N$, and the *block leverage scores* [**?**], [**?**] are defined as $\tilde{\ell}_\iota := \|\tilde{\mathbf{U}}_{(\mathcal{K}_\iota)}\|_F^2 = \sum_{j \in \mathcal{K}_\iota} \ell_j$ for all $\iota \in \mathbb{N}_K$. A lot of work has been done regarding ℓ_2-s.e. by leverage score sampling [**?**], [**?**], [**?**], [**?**], [**?**] as an importance sampling technique. By generalizing these to sampling blocks, one can show analogous results (*e.g.* [**?**], [**?**]).

Lemma **??** suggests that the *normalized* block leverage scores $\grave{\ell}_i = \frac{\tilde{\ell}_i}{K}$ of $\tilde{\mathbf{U}}$ are approximately uniform for all ι with high probability. This is the key step to proving that each $\mathbf{S}_{\boldsymbol{\Pi}}^{[t]}$ of Algorithm **??**, satisfy (**??**). We illustrate the flattening of the scores for the various random projections considered in this paper, in Figure **??**.

Lemma 3. *For all $\iota \in \mathbb{N}_K$ and $\mathcal{K}_\iota \subsetneq \mathbb{N}_N$ of size $\tau = N/K$*

$$\Pr\left[\grave{\ell}_\iota <_{N\rho} 1/K\right] = \Pr\left[\left|\grave{\ell}_\iota - 1/K\right| < \tau\rho\right] > 1 - \delta,$$

for $\rho \geqslant \sqrt{\log(2\tau/\delta)/2}$.

Theorem 2. *Fix $\epsilon > 0$ such that $\epsilon \ll 1/N$. By Lemma **??**, we can then assume that $\grave{\ell}_\iota = 1/K$ for all $\iota \in \mathbb{N}_K$. Then, $\mathbf{S}_{\boldsymbol{\Pi}}$ of Algorithm **??** is a ℓ_2-s.e. sketching matrix of $\mathbf{A}$, according to (**??**). Specifically, for $\delta > 0$ and $q = \Theta\left(\frac{d}{\tau}\log(2d/\delta)/\epsilon^2\right)$:*

$$\Pr\left[\|\mathbf{I}_d - \mathbf{U}^T \mathbf{S}_{\boldsymbol{\Pi}}^T \mathbf{S}_{\boldsymbol{\Pi}} \mathbf{U}\|_2 \leqslant \epsilon\right] \geqslant 1 - \delta.$$

To prove Lemma **??** we use Hoeffding's inequality to show that the individual leverage scores are flattened, and then group them together by applying the binomial approximation. This is then directly applied to a generalized version of the leverage score sketching matrix which samples blocks instead of individual rows [**?**, Theorem 1], to prove Theorem **??**.
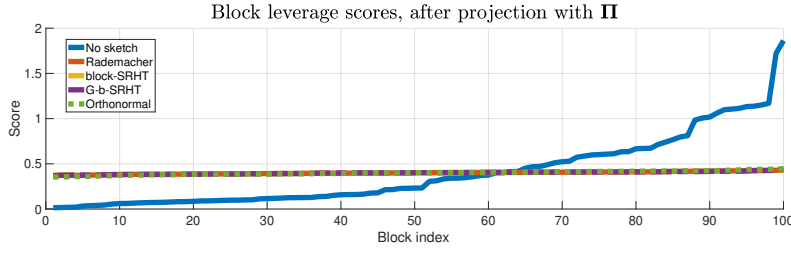
Fig. 2. Flattening of block-scores, for $\mathbf{A}$ following a t-distribution. We abbreviate the garbled block-SRHT to 'G-b-SRHT'.

We note that there is no benefit in considering an overlap across the block batches which are sent to the workers (*e.g.* if a worker receives $\begin{bmatrix} \tilde{\mathbf{A}}_{i-1}^T & \tilde{\mathbf{A}}_i^T \end{bmatrix}^T$ and another receives $\begin{bmatrix} \tilde{\mathbf{A}}_i^T & \tilde{\mathbf{A}}_{i+1}^T \end{bmatrix}^T$), in terms of sampling. The reason is that since the computations are received uniformly at random, there is still the same chance that $\hat{g}_i$ and $\hat{g}_j$ would be considered, for any $i \neq j$.

Before moving onto the block-SRHT, we show how our scheme compares to other approximate GCSs in terms of the approximation error (**??**), when we consider multiple replications of each encoded block being shared among the workers. The result of Proposition **??** also applies to other sketching approaches, which satisfy (**??**).

Proposition 1. *By Theorem* **??**, $\mathbf{S}_{\hat{\mathbf{\Pi}}}$ *satisfies* (**??**) *(w.h.p.). Hence, the approximate gradients* $\hat{g}^{[t]}$ *of Algorithm* **??** *satisfy* (**??**) *(w.h.p.), with* $\mathrm{err}(\mathbf{G}_{(\mathcal{I})}) = \epsilon/\sqrt{K}$.

IV. THE BLOCK-SRHT

In this section, we focus on a special case of $\mathbf{\Pi}$ which can be utilized in Algorithm **??**; the *randomized Hadamard transform*. By utilizing this transform we satisfy property (g), and also avoid the extra computational cost which is needed to generate a random orthonormal matrix [**?**].

The SRHT is comprised of three matrices: $\mathbf{\Omega} \in \mathbb{R}^{r \times N}$ a uniform sampling and rescaling matrix of r rows, $\hat{\mathbf{H}}_N \in \{\pm 1/\sqrt{N}\}^{N \times N}$ the normalized Hadamard matrix for $N = 2^n$, and $\mathbf{D} \in \{0, \pm 1\}^{N \times N}$ with i.i.d. diagonal Rademacher random entries; *i.e.* it is a signature matrix. The main intuition of the projection is that it expresses the original signal or feature-row in the Walsh-Hadamard basis. Furthermore, $\hat{\mathbf{H}}_N$ can be applied efficiently due to its structure. As in the case where we transformed the left orthonormal basis and column-space of $\mathbf{A}$ by multiplying its columns with a random orthonormal matrix $\mathbf{\Pi}$, in the new basis $\hat{\mathbf{H}}_N \mathbf{D} \mathbf{U}$; the block leverage scores are close to uniform. Hence, we perform uniform sampling through $\widehat{\mathbf{\Omega}}$ on the blocks of $\hat{\mathbf{H}}_N \mathbf{D} \mathbf{A}$ to reduce the effective dimension N, whilst the information of $\mathbf{A}$ is maintained.

To exploit the SRHT in distributed GC for linear regression, we generalize it to subsampling blocks instead of rows; of the transformed data matrix, as in Algorithm **??**. We give a ℓ_2-s.e. guarantee for the block-wise sampling version or SRHT, which characterizes the approximation of our proposed GCS for linear regression.

We refer to this special case as the "block-SRHT", for which $\hat{\mathbf{\Pi}}$ is taken from the subset $\hat{H}_N$ of $O_N(\mathbb{R})$

$$\hat{H}_N := \left\{ \hat{\mathbf{H}}_N \mathbf{D} : \mathbf{D} = \mathrm{diag}(\pm 1) \in \{0, \pm 1\}^{N \times N} \right\}, \tag{12}$$

where $\mathbf{D}$ is a random signature matrix with equiprobable entries of +1 and -1, and $\hat{\mathbf{H}}_N$ for $N = 2^n$ is defined by

$$\mathbf{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \hat{\mathbf{H}}_N = \frac{1}{\sqrt{N}} \cdot \mathbf{H}_2^{\otimes \log_2(N)} \ .$$

Equivalently, $\hat{\mathbf{H}}_N$ can be defined entry-wise through the n-bit binary representation of i,j as

$$\hat{\mathbf{H}}_{ij} = (-1)^{\langle i,j \rangle_2}/\sqrt{N} \ \text{ for } \ \langle i,j \rangle_2 = \left(\sum_{l=0}^{n-1} i_l \cdot j_l \right) \bmod 2 \ .$$

The SRHT introduced in [**?**] corresponds to the case where we select $\tau = 1$, *i.e.* $K = N$. Henceforth, we drop the subscript N.

The main differences between the SRHT and the proposed block-SRHT $\mathbf{S}_{\tilde{\mathbf{\Pi}}}$ for $\tilde{\mathbf{\Pi}} \overset{U}{\leftarrow} \hat{H}_N$, is the sampling matrix $\widetilde{\mathbf{\Omega}}$; and that $q = r/\tau$ sampling trials take place instead of r. The limiting computational step of applying $\mathbf{S}_{\hat{\mathbf{\Pi}}}$ in (**??**) is the multiplication by $\hat{\mathbf{H}}$. The recursive structure of $\hat{\mathbf{H}}$ permits us to compute $\mathbf{S}_{\hat{\mathbf{\Pi}}} \mathbf{A}$ in $\mathcal{O}(Nd \log N)$ time, through Fourier methods [**?**].

A. Subspace Embedding of the Block-SRHT

To show that $\mathbf{S}_{\hat{\mathbf{\Pi}}}$ with $\hat{\mathbf{\Pi}} \overset{U}{\leftarrow} \hat{H}_N$ satisfies (**??**), we first present a key result, analogous to that of Lemma **??**. Considering the orthonormal basis $\hat{\mathbf{V}} := \hat{\mathbf{H}} \mathbf{D} \mathbf{U}$ of the transformed data $\hat{\mathbf{H}} \mathbf{D} \mathbf{A}$ with individual leverage scores $\{\ell_i\}_{i=1}^N$, Lemma **??** suggests that the resulting block leverage scores $\tilde{\ell}_\iota = \|\hat{\mathbf{V}}_{(\mathcal{K}_\iota)}\|_F^2 = \sum_{j \in \mathcal{K}_\iota} \ell_j$ are approximately uniform for all $\iota \in \mathbb{N}_K$. Note that the diagonal

entries of $\mathbf{D}$ is the only place in which randomness takes place other than the sampling. This then allows us to prove our ℓ_2-s.e. result regarding the block-SRHT, Theorem **??**.

Lemma 4. *For all $\iota \in \mathbb{N}_K$ and $\mathcal{K}_\iota \subsetneq \mathbb{N}_N$ of size $\tau = N/K$*

$$\Pr\left[\tilde{\ell}_\iota \leqslant \eta d \cdot \log(Nd/\delta)/K\right] > 1 - \tau\delta/2, \tag{13}$$

for $0 < \eta \leqslant 2 + \log(16)/\log(Nd/\delta)$ a constant.

Theorem 3. *The block-SRHT $\mathbf{S}_{\hat{\mathbf{\Pi}}}$ is a ℓ_2-s.e. of $\mathbf{A}$. For $\delta > 0$ and $q = \Theta\left(\frac{d}{\tau}\log(Nd/\delta) \cdot \log(2d/\delta)/\epsilon^2\right)$:*

$$\Pr\left[\|\mathbf{I}_d - \mathbf{U}^T\mathbf{S}_{\hat{\mathbf{\Pi}}}^T\mathbf{S}_{\hat{\mathbf{\Pi}}}\mathbf{U}\|_2 \leqslant \epsilon\right] \geqslant 1 - \delta.$$

Compared to Theorem **??**, the above theorem has an additional logarithmic dependence on N. This is a consequence of applying the union bound, in order to show that the leverage scores of $\hat{\mathbf{H}}\mathbf{D}\mathbf{U}$ are flattened (Lemma **??**). In the proof of Lemma **??**, we instead applied Hoeffding's inequality, which removes such conditioning. Since Lemma **??** also holds for the block-SRHT, $\mathbf{S}_{\hat{\mathbf{\Pi}}}$ also satisfies the ℓ_2-s.e. guarantee stated in Theorem **??**.

In Subsection **??** we alter the transformation $\hat{\mathbf{H}}\mathbf{D}$ by permuting its rows. While our ℓ_2-s.e. result remains intact, under mild but necessary assumptions, this transformation now also guarantees computational security.

B. Recursive Kronecker Product Orthogonal Matrices

One could consider more general sets of matrices to sample from, while still benefiting from the recursive structure leveraged in Fourier methods. For a fixed 'base dimension' of $k \in \mathbb{Z}_{>2}$, let $\mathbf{\Pi}_k \in O_k(\mathbb{R})$, and define $\mathbf{\Pi} = \mathbf{\Pi}_k^{\otimes\lceil\log_k(N)\rceil}$. Carrying out the multiplication $\mathbf{\Pi}\mathbf{A}$ now takes $\mathcal{O}(Ndk^2\log_k N)$ time.

In the case where $k = 2$, up to a permutation of the rows and columns; we have $O_2(\mathbb{R}) = \{\mathbf{I}_2, \hat{\mathbf{H}}_2\}$, which is limiting compared to $O_k(\mathbb{R})$ for $k \geqslant 3$. This allows more flexibility, as more 'base matrices' $\mathbf{\Pi}_k$ can be considered, and the security can therefore be improved, as now we do not rely only on applying a random permutation to $\hat{\mathbf{\Pi}}$ (discussed in **??**).

V. OPTIMAL STEP-SIZE AND ADAPTIVE GC

Recently, *adaptive gradient coding* (AGC) has been proposed in [**?**]. The objective is to adaptively design an exact GCS without prior information about the behavior of potential persistent stragglers, in order to gradually minimize the communication load. This though comes at the cost of further delays due to intermediate designs of GC encoding-decoding pairs, as well as performing the encoding and decoding steps. Furthermore, the assumptions made in [**?**] are more stringent compared to the ones we have made thus far.

In this section, we further speed up our process, by adaptively selecting a step-size which reduces the total number of iterations required for convergence to the solutions of problems (**??**), (**??**) and (**??**), when SD is carried out. The proposed choice $\xi_t^\star$ for the step-size, is based on the latest gradient update of (**??**) and (**??**). To determine $\xi_t^\star$, we solve

$$\xi_t^\star = \arg\min_{\xi\in\mathbb{R}_{\geqslant 0}}\left\{\|\mathbf{A}\mathbf{x}^{[t+1]} - \mathbf{b}\|_2^2\right\} = \arg\min_{\xi\in\mathbb{R}_{\geqslant 0}}\left\{\|\mathbf{A}(\mathbf{x}^{[t]} - \xi\cdot g^{[t]})\mathbf{b}\|_2^2\right\} \tag{14}$$

for each t. If $\xi_t = 0$, we have reached the global optimum.

Since (**??**) has a closed form solution, determining $\xi_t^\star$ at each iteration reduces to matrix-vector multiplications. In the distributive setting, this will be determined by the central server once sufficiently many workers have responded at iteration t, who will then update $\mathbf{x}^{[t+1]}$ according to (**??**).

Compared to AGC, this is a more practical model, as we do not design and deploy multiple codes across the network. The authors of [**?**] minimize the communication load of individual communication rounds. In contrast, we reduce the total number of iterations of the SD procedure, which leads to fewer communication rounds. Depending on the application and threshold parameters we pick for the two respective AGC methods, our proposed approach would most likely have a lower overall communication load. This of course would also depend on the selected step-size used in the AGC for [**?**], and termination criterion. Furthermore, we are also flexible in tolerating a different number of stragglers s at each iteration, which was a motivation for the design of AGC schemes.

Proposition 2. *Given the respective gradient $g^{[t]}$ and update $\mathbf{x}^{[t]}$ of the underlying objective function, the optimal step-size according to (**??**) for $L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$, $L_{\mathbf{G}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$ and $L_{\mathbf{\Pi}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]}) := \|\mathbf{\Pi}(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2^2$, is:*

$$\xi_t^\star = \langle\mathbf{A}g^{[t]}, \mathbf{A}\mathbf{x}^{[t]} - \mathbf{b}\rangle/\|\mathbf{A}g^{[t]}\|_2^2. \tag{15}$$

In our distributive stochastic procedure, one could select an adaptive step-size ξ_{t+1} which minimizes $L_{\mathbf{S}_{\mathbf{\Pi}}^{[t]}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$; but the induced sketching matrix $\mathbf{S}_{\mathbf{\Pi}}^{[t]}$ would need to be explicitly determined once q workers have responded. This would result in further computations from the central server. Instead, we propose using the step-size (**??**), as it is optimal in expectation.

The bottleneck in using $\xi_t^\star$, is that it can only be updated once the $\tilde{g}^{[t]} := \nabla_{\mathbf{x}} L_{\mathbf{\Pi}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$ has been determined, which causes a delay in updating $\hat{\mathbf{x}}^{[t+1]}$. Even so, we significantly reduce the number of iterations, which is evident through our experiments in Section **??**. The overall computation of the entire network is therefore also reduced. Furthermore, $\mathbf{A}^T \mathbf{A}$ and $\mathbf{A}^T \mathbf{A} \mathbf{b}$ which appear in the expansion of (**??**) can be computed beforehand, so that $\mathbf{A}^T \mathbf{A} \mathbf{x}^{[t]}$ can be calculated by the central server while the workers are carrying out their tasks.

Corollary 2. *Assume that we know the parameter update $\hat{\mathbf{x}}^{[t]}$, and the gradient $\tilde{g}^{[t]}$. Over the possible index sets $\mathcal{S}^{[t]}$ at iteration t, the optimal step-size according to*

$$\arg \min_{\xi \in \mathbb{R}} \left\{ \mathbb{E} \left[\left\| \mathbf{S}_{\mathbf{\Pi}}^{[t]} \left(\mathbf{A} \hat{\mathbf{x}}^{[t+1]} - \mathbf{b} \right) \right\|_2^2 \right] \right\}$$

*matches $\xi_t^\star$ of (**??**).*

VI. SECURITY OF ORTHONORMAL SKETCHES

In this section, we discuss the security of the proposed orthonormal-based sketching matrices and the block-SRHT. The idea behind securing the resulting sketches is that there is a large ensemble of orthonormal matrices $\mathbf{\Pi}$ to select from, making it near-impossible for adversaries to discover the inverse transformation.

To give information-theoretic security guarantees, we make some mild but necessary assumptions regarding Algorithm **??** and the data matrix $\mathbf{A}$. The message space $\mathcal{M}$ needs to be finite, which $\mathcal{M}$ in our case corresponds to the set of possible orthonormal bases of the column-space of $\mathbf{A}$. This is something we do not have control over, and it depends on the application and distribution from which we assume the data is gathered. Therefore, we assume that $\mathcal{M}$ is finite. For this reason, we consider a finite multiplicative subgroup $(\tilde{O}_{\mathbf{A}}, \cdot)$ of $O_N(\mathbb{R})$ (thus $\mathbf{I}_N \in \tilde{O}_{\mathbf{A}}$, and if $\mathbf{Q} \in \tilde{O}_{\mathbf{A}}$ then $\mathbf{Q}^T \in \tilde{O}_{\mathbf{A}}$), which contains all potential orthonormal bases of $\mathbf{A}$.[3] Recall that $O_N(\mathbb{R})$ is a regular submanifold of $\mathrm{GL}_N(\mathbb{R})$. Hence, we can define a distribution on any subset of $O_N(\mathbb{R})$.

We then let $\mathcal{M} = \tilde{O}_{\mathbf{A}}$, and assume $\mathbf{U}_{\mathbf{A}}$ the $N \times N$ orthonormal basis of $\mathbf{A}$ is drawn from $\mathcal{M}$ w.r.t. D. For simplicity, we consider D to be the uniform distribution. A simple method of generating a random matrix that follows the uniform distribution on the Stiefel manifold $V_n(\mathbb{R}^n)$ can be found in [**?**, Theorem 2.2.1]. Alternatively, one could generate a random Gaussian matrix and then perform Gram–Schmidt in order to orthonormalize it. Furthermore, an inherent limitation of Shannon secrecy is that $|\mathcal{K}| \geqslant |\mathcal{M}|$.

Theorem 4. *In Algorithm **??**, sample $\mathbf{\Pi}$ uniformly at random from $\tilde{O}_{\mathbf{A}}$. The application of $\mathbf{\Pi}$ to $\mathbf{A}$ before partitioning the data, provides Shannon secrecy to $\mathbf{A}$ w.r.t. D uniform, for $\mathcal{K}, \mathcal{M}, \mathcal{C}$ all equal to $\tilde{O}_{\mathbf{A}}$.*

A. Securing the SRHT

Unfortunately, the guarantee of Theorem **??** does not apply to the block-SRHT, as in this case it is restrictive to assume that $\mathbf{U}_{\mathbf{A}} \in \hat{H}_N$. A simple computation on a specific example also shows that this sketching approach does not provide Shannon secrecy.[4] For instance, if $\mathbf{U}_0 = \mathbf{I}_2$, $\mathbf{U}_1 = \hat{\mathbf{H}}_2$ and the observed transformed basis $\bar{\mathbf{C}}$ has two zero entries, then

$$\Pr_{\mathbf{\Pi} \xleftarrow{U} H_2} \left[\mathbf{\Pi} \cdot \mathbf{U}_1 = \bar{\mathbf{C}} \right] > \Pr_{\mathbf{\Pi} \xleftarrow{U} H_2} \left[\mathbf{\Pi} \cdot \mathbf{U}_0 = \bar{\mathbf{C}} \right] = 0.$$

Furthermore, since $\hat{\mathbf{H}}$ is a known orthonormal matrix, it is a trivial task to invert this projection and reveal $\mathbf{DA}$. This shows that the inherent security of the SRHT is relatively weak. Proposition **??** is proven by constructing a counterexample.

Proposition 3. *The SRHT does not provide Shannon secrecy.*

To secure the SRHT and the block-SRHT, we randomly permute the rows of $\hat{\mathbf{H}}$, before applying it to $\mathbf{A}$. That is, for $\mathbf{P} \in S_N$ where $S_N \subsetneq \{0,1\}^{N \times N}$ is the permutation group on $N \times N$ matrices, we let $\tilde{\mathbf{H}} := \mathbf{P}\hat{\mathbf{H}} \in \{\pm 1/\sqrt{N}\}^{N \times N}$, and the new sketching matrix is

$$\mathbf{S}_{\tilde{\mathbf{\Pi}}} = \widetilde{\mathbf{\Omega}} \cdot (\mathbf{P} \cdot \hat{\mathbf{H}}) \cdot \mathbf{D} = \widetilde{\mathbf{\Omega}} \cdot \tilde{\mathbf{H}} \cdot \mathbf{D} = \widetilde{\mathbf{\Omega}} \cdot \tilde{\mathbf{\Pi}}, \tag{16}$$

for which our flattening result (Corollary **??**) still holds. We "garble" $\hat{\mathbf{H}}$ so that the projection applied to $\mathbf{A}$ now inherently has more randomness, and allows us to draw from a larger ensemble. Specifically, for a fixed N, the block-SRHT has 2^N options for $\hat{\mathbf{\Pi}} = \hat{\mathbf{H}}\mathbf{D}$, while for $\tilde{\mathbf{\Pi}} = \tilde{\mathbf{H}}\mathbf{D}$ there are $2^N N! = \mathcal{O}\big((2N/e)^N \sqrt{N}\big)$ options for $\tilde{\mathbf{\Pi}}$. Moreover, for

$$\tilde{H}_N := \left\{ \mathbf{P}\mathbf{\Pi} : \mathbf{P} \in S_N \text{ and } \mathbf{\Pi} \in \hat{H}_N \right\} \tag{17}$$

the set of all possible *garbled Hadamard transforms*, it follows that $(\tilde{H}_N, \cdot)$ is a finite multiplicative subgroup of $O_N(\mathbb{R})$. Hence, we can also define a distribution on $\tilde{H}_N$. We also get the benefits of permuting $\hat{\mathbf{H}}$'s columns without explicitly applying a second permutation, through $\mathbf{D}$.

---

[3] In Appendix **??**, we give an analogy between our approach and the OTP.
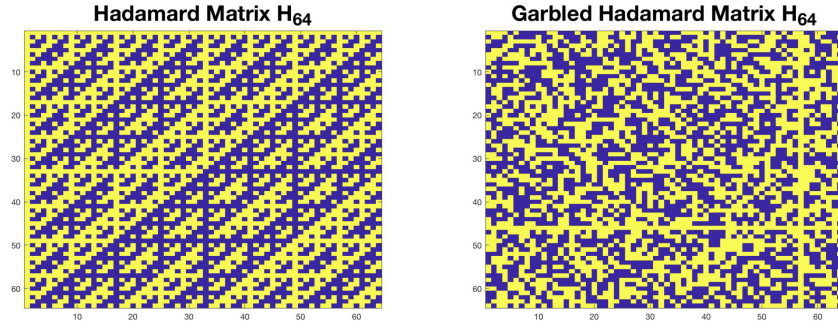[4] Please check Appendix **??** for the details.

Fig. 3. Example of how $\mathbf{P}$ and $\mathbf{D}$ modify the projection matrix $\hat{\mathbf{H}}_{64}$.

By the following Corollary, we deduce that Theorem **??** also holds for the "*garbled block-SRHT*" (an analogous result is used to prove Lemma **??**). Thus, we can apply any $\tilde{\mathbf{\Pi}} \xleftarrow{U} \tilde{H}_N$ in Algorithm **??**, and get a valid sketch.

Corollary 3. *For* $\mathbf{y} \in \mathbb{R}^N$ *a fixed (orthonormal) column vector of* $\mathbf{U}$, *and* $\mathbf{D} \in \{0, \pm 1\}^{N \times N}$ *with random equi-probable diagonal entries of* ± 1*, we have:*

$$\Pr\left[\|\tilde{\mathbf{H}}\mathbf{D} \cdot \mathbf{y}\|_\infty > C\sqrt{\log(Nd/\delta)/N}\right] \leqslant \frac{\delta}{2d} \tag{18}$$

for $0 < C \leqslant \sqrt{2 + \log(16)/\log(Nd/\delta)}$ *a constant.*

Moreover, Corollary **??** also holds true for random projections $\mathbf{R}$ whose entries are rescaled Rademacher random variables, *i.e.* $\mathbf{R}_{ij} = \pm 1/\sqrt{N}$ with equal probability. The advantage of this is that we have a larger set of projections

$$\tilde{R}_N := \left\{\mathbf{R} \in \{\pm 1/\sqrt{N}\}^{N \times N} : \Pr[\mathbf{R}_{ij} = +1/\sqrt{N}] = 1/2\right\}$$

to draw from. This makes it even harder for an adversary to determine which projection was applied. Specifically $|\tilde{R}_N| = 2^{N^2}$, which is significantly larger than $|\tilde{H}_N|$. Two drawbacks of applying a random Rademacher projection $\mathbf{R}$ is that it is much slower than its Hadamard-based counterpart, and the resulting gradients $\hat{g}^{[t]}$ are not unbiased.

Next, we provide a computationally secure guarantee for the garbled block-SRHT $\mathbf{S}_{\tilde{\mathbf{\Pi}}} \leftarrow \widetilde{\mathbf{\Omega}}\tilde{\mathbf{\Pi}}$. The guarantee of Theorem **??** against computationally bounded adversaries, relies heavily on the assumption that *strong pseudorandom permutations* (s-PRPs) and *one-way functions* (OWFs) exist. Through a long line of work, it was shown that s-PRPs exist if and only if OWFs exist. Even though OWFs are minimal cryptographic objects, it is not known whether such functions exist [**?**]. Proving their existence is non-trivial, as this would then imply that $\mathsf{P} \neq \mathsf{NP}$. In practice however, this is not unreasonable to assume. The proof of Theorem **??** entails a reduction to inverting the s-PRP $\mathbf{P}$. In practice, *block ciphers* are used for s-PRPs.

Theorem 5. *Assume that* $\mathbf{P}$ *is a s-PRP. Then,* $\mathbf{S}_{\tilde{\mathbf{\Pi}}}\mathbf{A}$ *is computationally secure against polynomial-bounded adversaries, for* $\mathbf{S}_{\tilde{\mathbf{\Pi}}} \leftarrow \widetilde{\mathbf{\Omega}}\tilde{\mathbf{\Pi}}$ *the garbled block-SRHT.*

As discussed in **??**, the Hadamard matrix satisfies the desired properties (b), (c), (d), (g), while any other form of a discrete Fourier transform would violate (c). By applying $\mathbf{P}$ to $\hat{\mathbf{H}}$, the matrix $\mathbf{P}\hat{\mathbf{H}}$ still satisfies the aforementioned properties, while also incorporating security; *i.e.* property (f). It would be interesting to see if other structured matrices exist which also satisfy (b)-(g). Similar to what we saw with the block-SRHT, if (b) is met; then we can achieve (a) through uniform sampling.

B. Exact Gradient Recovery

In the case where the *exact* gradient is desired, one can use the proposed orthonormal projections to encrypt the information from the workers, while requiring that the computations from *all* the workers are received. From Theorems **??** and **??**, we know that under certain assumptions we can secure $\mathbf{A}$.

Since the projections are orthonormal, it would follow that $\hat{g}^{[t]} = g_{ls}^{[t]}$. Thus, as long as *all* workers respond, the aggregated gradient is equal to the exact gradient. One can utilize this idea to encrypt other distributive computations, *e.g.* matrix multiplication or inversion and logistic regression, which are discussed in Appendix **??**. This resembles a homomorphic encryption scheme, but is by no means fully-homomorphic.

VII. EXPERIMENTS

We compared our proposed distributed GCSs to analogous approaches where the projection $\mathbf{\Pi}$ is a Gaussian sketch or a Rademacher random matrix. Our approach was found to outperform both these sketching methods in terms of convergence and approximation error, as the resulting gradients through these alternative approaches are not unbiased. In all experiments, the same initialization $\mathbf{x}^{[0]}$ was selected for each sketching methods.

Our approach was also compared to uncoded (regular) SD. Random matrices $\mathbf{A} \in \mathbb{R}^{2000 \times 40}$ with non-uniform block leverage scores were generated for the experiments. Standard Gaussian noise was added to an arbitrary vector from $\mathrm{im}(\mathbf{A})$, to define $\mathbf{b}$. We considered $K = 100$ blocks, thus $\tau = 20$. The effective dimension N was reduced to $r = 1000$.

For the experiments in Figure **??** we ran 600 iterations on six different instances for each one, and varied ξ for each experiment by logarithmic factors of the step-size $\xi^{\times} = 2/\sigma_{\max}(\mathbf{A})^2$. The average $\log$ residual errors $\log_{10}\left(\|\mathbf{x}_{ls}^{\star} - \hat{\mathbf{x}}\|_2/\sqrt{N}\right)$ are depicted in Figure **??**. Step-size $\xi^{\times}$ was considered, as it guarantees descent at each iteration, though it is too conservative.
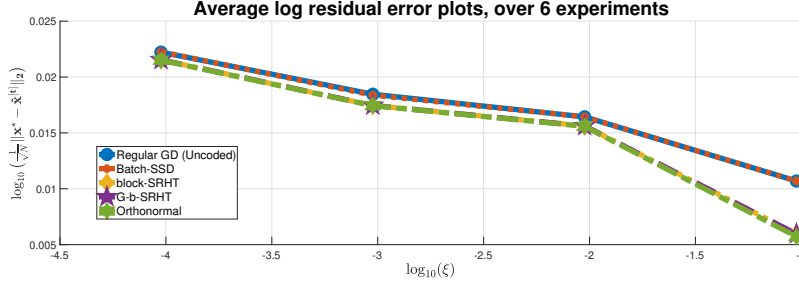

Fig. 4. log residual error, for $\mathbf{A}$ following a t-distribution.

In contrast to the Gaussian sketch, orthonormal matrices $\mathbf{\Pi}$ also act as preconditioners. One example is the experiment depicted in Figure **??**, in which the only modification we made from the previous experiments, was our initial choice of $\mathbf{x}^{[0]}$, which was scaled by $1/N$.
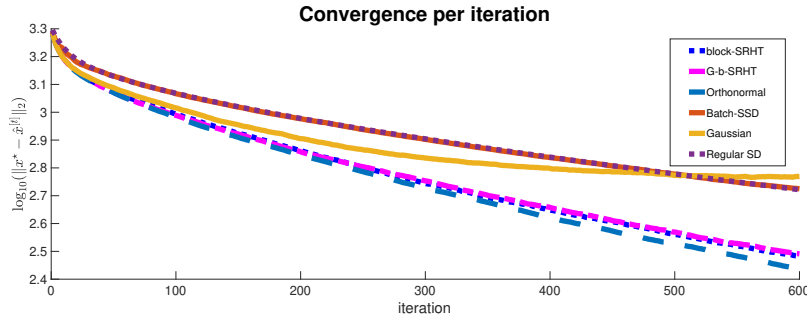

Fig. 5. Example where $\mathbf{\Pi}$ also acts as a preconditioner.

Next, we consider the case where ξ_t was updated according to (**??**). As above, our sketching approaches outperformed the case where a Gaussian sketch was applied. From Figure **??**, our orthonormal sketching approach performs just as well as regular SD for the first 30 iterations, though it slows down afterwards, and is slightly worse than regular SD by the time 50 iterations have been completed. By the discussion in **??**, we can achieve the performance of regular SD if we wait until all workers respond; and consider no stragglers, while our security guarantees still hold. This is true also for the block-SRHT and garbled block-SRHT, but not for the Gaussian sketch.
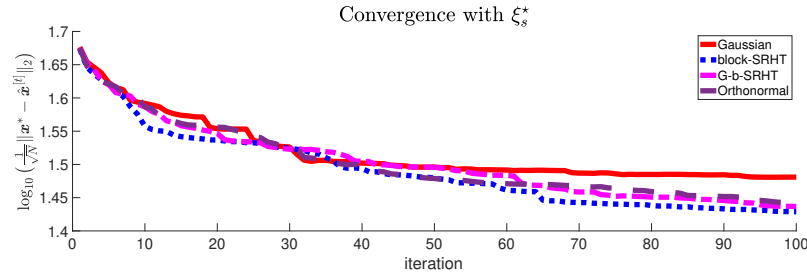
Fig. 6. Adaptive step-size update, for $\mathbf{A}$ following a t-distribution.

Lastly, we give an example where it is clear that iterative sketching leads to better convergence than the sketch-and-solve approach. In the experiment depicted in Figure **??**, we considered three sketching approaches: the iterative block-SRHT and garbled block-SRHT, and the non-iterative garbled block-SRHT. The step-size was adaptive at each iteration, as was done in the experiment of Figure **??**.

We carried out similar experiments when considering other dense and sparse matrices $\mathbf{A}$, with non-uniform block leverage scores. Similar results regarding our approaches were observed, as the ones provided above.

Fig. 7. Convergence at each step, for the iterative block-SRHT and garbled block-SRHT, and the non-iterative garbled block-SRHT.

VIII. CONCLUDING REMARKS AND FUTURE WORK

In this work, we proposed approximately solving a linear system by distributively leveraging iterative sketching and performing first-order SD simultaneously. In doing so, we 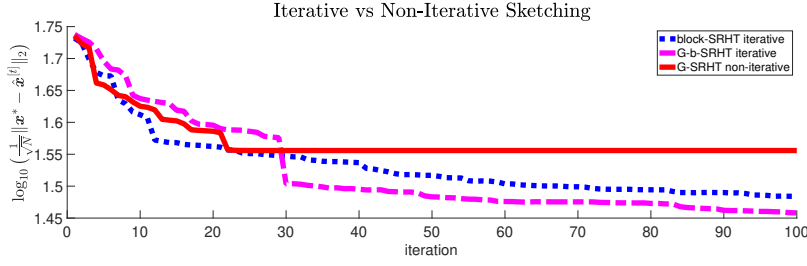benefit from both approximate GC and RandNLA. A difference to other RandNLA works is that our sketching matrices sample *blocks* uniformly at random, after applying a random orthonormal projection. An additional benefit is that by considering a large ensemble of orthonormal matrices to pick from, under necessary assumptions, we guarantee information-theoretic security while performing the distributed computations. This approach also enables us to not require encoding and decoding steps at every iteration. We also studied the special case where the projection is the randomized Hadamard transform, and discussed its security limitation. To overcome this, we proposed a modified "garbled block-SRHT", which guarantees computational security.

We note that applying orthonormal random matrices also secures coded matrix multiplication. There is a benefit when applying a garbled Hadamard transform in this scenario, as the complexity of multiplication resulting from the sketching is less than that of regular multiplication. Also, if such a random projection is used before performing CR-multiplication distributively [?], [?], [?], the approximate result will be the same. Moreover, our dimensionality reduction algorithm can be utilized by a single server, to store low-rank approximations of very large data matrices.

Partial stragglers, have also been of interest in the GC literature. These are stragglers who are able to send back a portion of their requested tasks. Our work is directly applicable, as we can consider smaller blocks, with multiple ones allocated to each worker.

There are several interesting directions for future work. We observed experimentally in Figure ?? that $\mathbf{\Pi}$ and $\hat{\mathbf{H}}_N$ may act as preconditioners for SSD. This mere observation requires further investigation. Another direction is to see if the proposed ideas could be applied to federated learning scenarios, in which security and privacy are major concerns. Some of the projections we considered, rely heavily on the recursive structure of $\hat{\mathbf{H}}$ in order to satisfy (g). One thing we overlooked, is whether other efficient multiplication algorithms (*e.g.* Strassen's [?]) could be exploited, in order to construct suitable projections. It would be interesting to see if other structured or sparse matrices exist which also satisfy our desired properties (a)-(g).

There has been a lot of work regarding second-order algorithms with iterative sketching, *e.g.* [?], [?]. Utilizing iterative Hessian sketching or sketched Newton's method in CC has been explored in a tangential work [?], though the security aspect of these algorithms has not been extensively studied. A drawback here is that the local computations at the workers would be much larger, though we expect the number of iterations to be significantly reduced; for the same termination criterion to be met, compared to first-order methods. Deeper exploration of the theoretical guarantees of iterative sketched first-order methods, along with a comparison to their second-order counterparts, as well as studying their effect in logistic regression and other applications, are also of potential interest.

APPENDIX A
PROOFS OF SECTION ??

A. Subsection ??

Note that in Lemma ??:

$$\mathbb{E}\left[\widetilde{\mathbf{\Omega}}_{[t]}^T \widetilde{\mathbf{\Omega}}_{[t]}\right] = \mathbf{I}_N \quad \Longrightarrow \quad \mathbb{E}\left[\mathbf{S}_{[t]}^T \mathbf{S}_{[t]}\right] = \mathbf{I}_N \ ,$$

as

$$\mathbb{E}\left[\mathbf{S}_{[t]}^T \mathbf{S}_{[t]}\right] = \mathbf{\Pi}^T \mathbb{E}\left[\widetilde{\mathbf{\Omega}}_{[t]}^T \widetilde{\mathbf{\Omega}}_{[t]}\right] \mathbf{\Pi} = \mathbf{\Pi}^T \mathbf{\Pi} = \mathbf{I}_N \ .$$

We provide both derivations separately in order to convey the respective importance behind the use of the Lemma in subsequent arguments, even though the main idea is the same. Furthermore, the proof of Theorem ?? is very similar to that of Lemma ??.

Proof. [Lemma **??**] The only difference in $\mathbf{S}_{\Pi}^{[t]}$ at each iteration, is $\mathcal{S}^{[t]}$ and $\widetilde{\boldsymbol{\Omega}}_{[t]}$. This corresponds to a uniform random selection of q out of K batches of the data which determine the gradient at iteration t — all blocks are scaled by the same factor $\sqrt{K/q}$ in $\widetilde{\boldsymbol{\Omega}}_{[t]}$. Let $\mathcal{Q}$ be the set of all subsets of $\mathbb{N}_K$ of size q. Then

$$
\begin{aligned}
\mathbb{E}\big[\mathbf{S}_{[t]}^T \mathbf{S}_{[t]}\big] &= \sum_{\mathcal{S}^{[t]}\in\mathcal{Q}} \frac{1}{\binom{K}{q}} \cdot \big(\mathbf{S}_{[t]} \cdot \mathbf{S}_{[t]}\big) \\
&= \frac{1}{\binom{K}{q}} \sum_{\mathcal{S}^{[t]}\in\mathcal{Q}} \sum_{i\in\mathcal{S}^{[t]}} \left(\sqrt{K/q}\right)^2 \cdot \boldsymbol{\Pi}_{(\mathcal{K}_i)}^T \boldsymbol{\Pi}_{(\mathcal{K}_i)} \\
&= \frac{\binom{K-1}{q-1}}{\binom{K}{q}} \sum_{i=1}^{K} \frac{K}{q} \cdot \boldsymbol{\Pi}_{(\mathcal{K}_i)}^T \boldsymbol{\Pi}_{(\mathcal{K}_i)} \\
&= \frac{\binom{K-1}{q-1}\cdot\frac{K}{q}}{\binom{K}{q}} \sum_{i=1}^{K} \boldsymbol{\Pi}_{(\mathcal{K}_i)}^T \boldsymbol{\Pi}_{(\mathcal{K}_i)} \\
&= \boldsymbol{\Pi}^T\boldsymbol{\Pi} \\
&= \mathbf{I}_N
\end{aligned}
$$

where $\binom{K-1}{q-1}$ is the number of sets in $\mathcal{Q}$ which include i, for each $i \in \mathbb{N}_K$. This completes the first part of the proof.

Note that the sampling and rescaling matrices $\widetilde{\boldsymbol{\Omega}}_{[t]}$ of Algorithm **??**, may also be expressed as

$$
\widetilde{\boldsymbol{\Omega}}_{[t]} = \sqrt{K/q}\cdot \sum_{\iota\in\mathcal{S}^{[t]}} \mathbf{I}_{(\mathcal{K}_\iota)} \ .
$$

Further notice that $\widetilde{\boldsymbol{\Omega}}_{[t]}$'s corresponding sampling and rescaling matrix of size $N\times N$, which appears in the expansion the objective function (**??**), is

$$
\begin{aligned}
\widetilde{\boldsymbol{\Omega}}_{[t]}^T \widetilde{\boldsymbol{\Omega}}_{[t]} &= \left(\sqrt{K/q}\right)^2 \cdot \sum_{\iota\in\mathcal{S}^{[t]}} \big(\mathbf{I}_{(\mathcal{K}_\iota)}\big)^T \mathbf{I}_{(\mathcal{K}_\iota)} \\
&= \frac{K}{q}\cdot \sum_{j\in\bigsqcup_{\iota\in\mathcal{S}^{[t]}}\mathcal{K}_\iota} \mathbf{e}_j\mathbf{e}_j^T \ .
\end{aligned}
$$

Let $\mathcal{B}$ denote the set of all possible block sampling and rescaling matrices of size $r \times N$, which sample q out of K blocks. For $\boldsymbol{\Phi} \in \mathcal{B}$, by $\mathbf{I}_{(\mathcal{K}_\iota)} \subseteq \boldsymbol{\Phi}$ we denote the condition that $\mathbf{I}_{(\mathcal{K}_\iota)}$ is a submatrix of $\boldsymbol{\Phi}$. Note that for each $\iota \in \mathbb{N}_K$, there are $\binom{K-1}{q-1}$ matrices in $\mathcal{B}$ which have $\mathbf{I}_{(\mathcal{K}_\iota)}$ as a submatrix. For our set up, we then have

$$
\begin{aligned}
\mathbb{E}\left[\widetilde{\boldsymbol{\Omega}}_{[t]}^T \widetilde{\boldsymbol{\Omega}}_{[t]}\right] &= \sum_{\boldsymbol{\Phi}\in\mathcal{B}} \frac{1}{\binom{K}{q}} \cdot \big(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\big) \\
&= \frac{\left(\sqrt{K/q}\right)^2}{\binom{K}{q}} \cdot \sum_{\boldsymbol{\Phi}\in\mathcal{B}} \sum_{\mathbf{I}_{(\mathcal{K}_\iota)}\subseteq\boldsymbol{\Phi}} \big(\mathbf{I}_{(\mathcal{K}_\iota)}\big)^T \mathbf{I}_{(\mathcal{K}_\iota)} \\
&= \frac{\binom{K-1}{q-1}\cdot(K/q)}{\binom{K}{q}} \cdot \sum_{\iota\in\mathbb{N}_K} \big(\mathbf{I}_{(\mathcal{K}_\iota)}\big)^T \mathbf{I}_{(\mathcal{K}_\iota)} \\
&= \sum_{\iota\in\mathbb{N}_K} \big(\mathbf{I}_{(\mathcal{K}_\iota)}\big)^T \mathbf{I}_{(\mathcal{K}_\iota)} \\
&= \sum_{j\in\mathbb{N}_N} \mathbf{e}_j^T\mathbf{e}_j \\
&= \mathbf{I}_N
\end{aligned}
$$

and the proof is complete.

$\square$

Proof. [Theorem **??**] The only difference in $\mathbf{S}_{\Pi}^{[t]}$ at each iteration, is $\mathcal{S}^{[t]}$ and $\widetilde{\boldsymbol{\Omega}}_{[t]}$. This corresponds to a uniform random selection of q out of K batches of the data which determine the gradient at iteration t — all blocks are scaled by the same factor $\sqrt{K/q}$ in $\widetilde{\boldsymbol{\Omega}}_{[t]}$. By (**??**), the gradient update is equal to that of a batch stochastic steepest descent procedure.

We break up the proof of the second statement by first showing that $\mathbb{E}\left[\hat{g}^{[t]}\right] = \tilde{g}^{[t]}$; for $\tilde{g}$ the gradient in the basis $\boldsymbol{\Pi}\mathbf{U}$, and then showing that $\mathbb{E}\left[\tilde{g}^{[t]}\right] = \frac{q}{K}\cdot g_{ls}^{[t]}$.

Let $\mathcal{Q}$ be the set of all subsets of $\mathbb{N}_K$ of size q, $\hat{g}_{\mathcal{S}^{[t]}}$ the gradient determined by the index set $\mathcal{S}^{[t]}$, and $\tilde{g}_i^{[t]}$ the respective partial gradients at iteration t. Then

$$
\begin{aligned}
\mathbb{E}\left[\hat{g}^{[t]}\right] &= \sum_{\mathcal{S}^{[t]} \in \mathcal{Q}} \frac{1}{\binom{K}{q}} \cdot \hat{g}_{\mathcal{S}^{[t]}} \\
&= \frac{1}{\binom{K}{q}} \sum_{\mathcal{S}^{[t]} \in \mathcal{Q}} \sum_{i \in \mathcal{S}^{[t]}} \left(\sqrt{K/q}\right)^2 \cdot \tilde{g}_i^{[t]} \\
&= \frac{\binom{K-1}{q-1}}{\binom{K}{q}} \sum_{i=1}^{K} \frac{K}{q} \cdot \tilde{g}_i^{[t]} \\
&= \sum_{i=1}^{K} \tilde{g}_i^{[t]} \\
&= \tilde{g}^{[t]}
\end{aligned}
$$

where $\binom{K-1}{q-1}$ is the number of sets in $\mathcal{Q}$ which include i, for each $i \in \mathbb{N}_K$.

We denote the resulting partial gradient on the sampled index set $\mathcal{S}^{[t]}$ of the gradient on (??) at iteration t; *i.e.* $g_{ls}^{[t]}$, by $g_{\mathcal{S}^{[t]}}$, and the individual partial gradients by $g_i^{[t]}$. Using the same notation as above, we get that

$$
\begin{aligned}
\mathbb{E}\left[\tilde{g}^{[t]}\right] &= \sum_{\mathcal{S}^{[t]} \in \mathcal{Q}} \frac{1}{\binom{K}{q}} \cdot g_{\mathcal{S}^{[t]}} \\
&= \frac{1}{\binom{K}{q}} \sum_{\mathcal{S}^{[t]} \in \mathcal{Q}} \sum_{i \in \mathcal{S}^{[t]}} g_i^{[t]} \\
&= \frac{\binom{K-1}{q-1}}{\binom{K}{q}} \sum_{i=1}^{K} g_i^{[t]} \\
&= \frac{q}{K} \cdot \sum_{i=1}^{K} \tilde{g}_i^{[t]} \\
&= \frac{q}{K} \cdot g^{[t]}
\end{aligned}
$$

which completes the proof. $\qquad\square$

Proof. [Lemma **??**] Since $\mathbf{\Pi}$ is an orthonormal matrix, the solution of the least squares problem with the objective $L_{\mathbf{G}}(\mathbf{A}, \mathbf{b}; \mathbf{x})$ is equal to the optimal solution (**??**), as

$$
\begin{aligned}
\hat{\mathbf{x}} &= \arg\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{G}(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2^2 \\
&= \arg\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{\Pi}(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2^2 \\
&= \arg\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\
&= \mathbf{x}_{ls}^{\star} .
\end{aligned}
$$

$\qquad\square$

Proof. [Corollary **??**] We prove this by induction. From our assumptions we have a fixed starting point $\mathbf{x}^{[0]}$, for which $\hat{\mathbf{x}}^{[0]} = \mathbf{x}^{[0]}$. Our base case is therefore $\mathbb{E}[\hat{\mathbf{x}}^{[0]}] = \mathbb{E}[\mathbf{x}^{[0]}] = \mathbf{x}^{[0]}$. For the inductive hypothesis, we assume that $\mathbb{E}[\hat{\mathbf{x}}^{[\tau]}] = \mathbf{x}^{[\tau]}$ for $\tau \in \mathbb{N}$.

It then follows that at step $\tau + 1$ we have

$$
\begin{aligned}
\mathbb{E}\left[\hat{\mathbf{x}}^{[\tau+1]}\right] &= \mathbb{E}\left[\hat{\mathbf{x}}^{[\tau]} - \hat{\xi}_\tau \cdot \hat{g}^{[\tau]}\right] \\
&= \mathbb{E}\left[\hat{\mathbf{x}}^{[\tau]}\right] - \frac{K}{q} \cdot \xi_\tau \cdot \mathbb{E}\left[\hat{g}^{[\tau]}\right] \\
&= \mathbf{x}^{[\tau]} - \frac{q}{K} \cdot \left(\frac{K}{q} \cdot \xi_\tau\right) \cdot g_{ls}^{[\tau]} \\
&= \mathbf{x}^{[\tau]} - \xi_\tau \cdot g_{ls}^{[\tau]} \\
&= \mathbf{x}^{[\tau+1]}
\end{aligned}
$$

which completes the inductive step. $\qquad\square$

B. Subsection **??**

In this appendix, we provide the proofs of Lemma **??** and Theorem **??**. First, we need to provide Lemmas **??** and **??**, and Hoeffding's inequality; which we use to prove the latter Lemma. Throughout this subsection, by ℓ_i we denote the ith leverage score of $\mathbf{\Pi A}$ for $\mathbf{\Pi}$ a random orthonormal matrix, *i.e.*

$$\ell_i = \|\tilde{\mathbf{U}}_{(i)}\|_2^2 = \|\mathbf{e}_i^T \tilde{\mathbf{U}}\|_2^2 = \mathbf{e}_i^T \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T \mathbf{e}_i \tag{19}$$

where $\tilde{\mathbf{U}} = \mathbf{\Pi U}$; for $\mathbf{U}$ the reduced left orthonormal matrix of $\mathbf{A}$. By $\mathbf{e}_i$ we denote the ith standard basis vector of $\mathbb{R}^N$.

Lemma 5. *For each $i \in \mathbb{N}_N$, we have $\mathbb{E}[\ell_i] = \frac{d}{N}$.*

Proof. By (**??**), we have

$$\begin{aligned}
\mathbb{E}[\ell_i] &= \mathbb{E}\left[\text{tr}(\mathbf{e}_i^T \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T \mathbf{e}_i)\right] \\
&= \mathbb{E}\left[\text{tr}(\mathbf{e}_i \mathbf{e}_i^T \cdot \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T)\right] \\
&= \sum_{j=1}^{N} \frac{1}{N} \cdot \text{tr}(\mathbf{e}_j \mathbf{e}_j^T \cdot \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T) \\
&= \frac{1}{N} \cdot \text{tr}\left(\sum_{j=1}^{N} \mathbf{e}_j \mathbf{e}_j^T \cdot \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T\right) \\
&= \frac{1}{N} \cdot \text{tr}\left(\mathbf{I}_N \cdot \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T\right) \\
&= \frac{1}{N} \cdot \text{tr}\left(\tilde{\mathbf{U}} \tilde{\mathbf{U}}^T\right) \\
&= \frac{d}{N} \ .
\end{aligned}$$

$\square$

Let $\bar{\ell}_i$ denote the ith normalized leverage score, *i.e.* $\bar{\ell}_i = \frac{\ell_i}{d}$. The ιth normalized block leverage score of $\mathbf{A}$ is denoted by $\grave{\ell}_\iota$, *i.e.*

$$\grave{\ell}_\iota = \frac{1}{d} \cdot \|\mathbf{I}_{(\mathcal{K}_\iota)} \tilde{\mathbf{U}}\|_F^2 = \frac{1}{d} \cdot \left(\sum_{j \in \mathcal{K}_\iota} \ell_j\right) = \sum_{j \in \mathcal{K}_\iota} \bar{\ell}_j \ . \tag{20}$$

To prove Lemma **??**, we first recall Hoeffding's inequality.

Theorem 6 (Hoeffding's Inequality, [**?**]). *Let $\{X_i\}_{i=1}^m$ be independent random variables such that $X_i \in [a_i, b_i]$ for all $i \in \mathbb{N}_m$, and let $X = \sum_{i=1}^m X_i$. Then*

$$\Pr\left[\left|X - \mathbb{E}[X]\right| \geqslant t\right] \leqslant 2 \cdot \exp\left\{\frac{-2t^2}{\sum_{j=1}^m (a_i - b_i)^2}\right\} \ .$$

Lemma 6. *The normalized leverage scores $\{\bar{\ell}_i\}_{i=1}^N$ of $\mathbf{\Pi A}$ satisfy*

$$\Pr\left[|\bar{\ell}_i - 1/N| < \rho\right] > 1 - 2 \cdot e^{-2\rho^2}$$

for any $\rho > 0$.

Proof. We know that $\ell_i \in [0, d]$ for each $i \in \mathbb{N}_N$, thus $\bar{\ell}_i \in [0, 1]$ for each i. By Lemma **??**, it follows that

$$\mathbb{E}[\bar{\ell}_i] = \mathbb{E}[\ell_i/d] = \frac{1}{d} \cdot \mathbb{E}[\ell_i] = \frac{1}{N}.$$

Now, fix a constant $\rho > 0$. By applying Theorem **??** with $m = 1$, we get

$$\Pr\left[|\bar{\ell}_i - 1/N| \geqslant \rho\right] \leqslant 2 \cdot e^{-2\rho^2}$$

thus

$$\Pr\left[|\bar{\ell}_i - 1/N| < \rho\right] > 1 - 2 \cdot e^{-2\rho^2}.$$

$\square$

Next, we complete the proof of the "flattening Lemma of block leverage scores" (Lemma **??**).

Proof. [Lemma **??**] To show that the two probability events of expression (**??**) are equal, note that:

1) $\grave{\ell}_\iota - \frac{1}{K} < \frac{N}{K}\rho \iff \grave{\ell}_\iota < (1 + N\rho)\frac{1}{K}$

2) $\frac{1}{K} - \grave{\ell}_\iota < \frac{N}{K}\rho \iff \grave{\ell}_\iota > (1 - N\rho)\frac{1}{K}$.

By combining the two inequalities, we conclude that

$$(1 - N\rho) \cdot \frac{1}{K} < \grave{\ell}_\iota < (1 + N\rho) \cdot \frac{1}{K} \iff \grave{\ell}_\iota <_{N\rho} 1/K. \tag{21}$$

By Lemma **??**, it follows that

$$\begin{aligned}
\Pr\left[\left|\grave{\ell}_\iota - 1/K\right| < \tau\rho\right] &> \Pr\left[\bigwedge_{j \in \mathcal{K}_\iota} \left\{|\bar{\ell}_i - 1/N| < \rho\right\}\right] \\
&> \left(1 - 2 \cdot e^{-2\rho^2}\right)^\tau \\
&\overset{\bowtie}{\approx} 1 - 2\tau \cdot e^{-2\rho^2}
\end{aligned}$$

where in $\bowtie$ we applied the binomial approximation. By substituting $\rho \geqslant \sqrt{\log(2\tau/\delta)/2}$, we get

$$\begin{aligned}
e^{-2\rho^2} &\leqslant e^{-2\frac{\log(2\tau/\delta)}{2}} \\
&= e^{-\log(2\tau/\delta)} \\
&= e^{\log(\delta/2\tau)} \\
&= \delta/2\tau,
\end{aligned}$$

thus $2\tau \cdot e^{-2\rho^2} \leqslant \delta$; and $1 - 2\tau \cdot e^{-2\rho^2} \geqslant 1 - \delta$. In turn, this implies that $\Pr\left[\left|\grave{\ell}_\iota - 1/K\right| < \tau\rho\right] > 1 - \delta$. $\qquad\square$

The proof of Theorem **??** is a direct consequence of Lemma **??** and Theorem **??**. In our statement we make the assumption that $\grave{\ell}_\iota = 1/K$ for all ι, though this is not necessarily the case, as Lemma **??** permits a small deviation. For $\rho \leftarrow \epsilon$, we consider $\epsilon \ll 1/N$ so that the '$N\epsilon$ multiplicative error' in (**??**) is small. We note that [**?**, Theorem 1] considers sampling according to *approximate* block leverage scores.

Theorem 7 (ℓ_2-s.e. of the block leverage score sampling sketch, [**?**]). *The sketching matrix $\widetilde{\mathbf{S}}$ constructed by sampling blocks of $\mathbf{A}$ with replacement according to their normalized block leverage scores $\{\grave{\ell}_\iota\}_{\iota=1}^K$ and rescaling each sampled block by $\sqrt{1/(q\grave{\ell}_\iota)}$, guarantees a ℓ_2-s.e. of $\mathbf{A}$; as defined in (**??**). Specifically, for $\delta > 0$ and $q = \Theta(\frac{d}{\tau}\log(2d/\delta)/\epsilon^2)$:*

$$\Pr\left[\|\mathbf{I}_d - \mathbf{U}^T\widetilde{\mathbf{S}}^T\widetilde{\mathbf{S}}\mathbf{U}\|_2 \leqslant \epsilon\right] \geqslant 1 - \delta.$$

Before we prove Proposition **??**, we first derive (**??**). In [**?**], the optimal decoding vector of an approximate GCS was defined as

$$\mathbf{a}_\mathcal{I}^\star = \arg\min_{\mathbf{a} \in \mathbb{R}^{1 \times q}} \left\{\|\mathbf{a}\mathbf{G}_{(\mathcal{I})} - \vec{\mathbf{1}}\|_2^2\right\}. \tag{22}$$

In the case where $q \geqslant K$, it follows that $\mathbf{a}_\mathcal{I}^\star = \vec{\mathbf{1}}\mathbf{G}_{(\mathcal{I})}^\dagger$. The error can then be quantified as

$$\mathrm{err}(\mathbf{G}_{(\mathcal{I})}) := \|\mathbf{I}_K - \mathbf{G}_{(\mathcal{I})}^\dagger\mathbf{G}_{(\mathcal{I})}\|_2.$$

The optimal decoding vector (**??**) has also been considered in other schemes, *e.g.* [**?**], [**?**].

Let $\mathbf{g}^{[t]}$ be the matrix comprised of the transposed exact partial gradients at iteration t, *i.e.*

$$\mathbf{g}^{[t]} := \left(g_1^{[t]} \quad g_2^{[t]} \quad \ldots \quad g_K^{[t]}\right)^T \in \mathbb{R}^{K \times d}. \tag{23}$$

Then, for a GCS $(\mathbf{G}, \mathbf{a}_\mathcal{I})$ satisfying $\mathbf{a}_\mathcal{I}\mathbf{G}_{(\mathcal{I})} = \vec{\mathbf{1}}$ for any $\mathcal{I}$, it follows that $(\mathbf{a}_\mathcal{I}\mathbf{G}_{(\mathcal{I})})\mathbf{g}^{[t]} = \vec{\mathbf{1}}\mathbf{g}^{[t]} = (g^{[t]})^T$. Hence, the gradient can be recovered exactly. Considering an optimal approximate scheme $(\mathbf{G}, \mathbf{a}_\mathcal{I}^\star)$ which recovers the gradient estimate $\grave{g}^{[t]} = (\mathbf{a}_\mathcal{I}^\star\mathbf{G}_{(\mathcal{I})})\mathbf{g}^{[s]}$, the error in the gradient approximation is

$$\begin{aligned}
\left\|g^{[s]} - \grave{g}^{[s]}\right\|_2 &= \left\|(\vec{\mathbf{1}} - \mathbf{a}_\mathcal{I}^\star\mathbf{G}_{(\mathcal{I})})\mathbf{g}^{[s]}\right\|_2 \\
&= \left\|\vec{\mathbf{1}}(\mathbf{I}_K - \mathbf{G}_{(\mathcal{I})}^\dagger\mathbf{G}_{(\mathcal{I})})\mathbf{g}^{[s]}\right\|_2 \\
&\leqslant \|\vec{\mathbf{1}}\|_2 \cdot \left\|\mathbf{I}_K - \mathbf{G}_{(\mathcal{I})}^\dagger\mathbf{G}_{(\mathcal{I})}\right\|_2 \cdot \left\|\mathbf{g}^{[s]}\right\|_2 \\
&\overset{\pounds}{\leqslant} \sqrt{K} \cdot \left\|\mathbf{I}_K - \mathbf{G}_{(\mathcal{I})}^\dagger\mathbf{G}_{(\mathcal{I})}\right\|_2 \cdot \left\|g^{[s]}\right\|_2 \\
&\overset{\$}{\leqslant} 2\sqrt{K} \cdot \underbrace{\left\|\mathbf{I}_K - \mathbf{G}_{(\mathcal{I})}^\dagger\mathbf{G}_{(\mathcal{I})}\right\|_2}_{\mathrm{err}(\mathbf{G}_{(\mathcal{I})})} \cdot \|\mathbf{A}\|_2 \cdot \|\mathbf{A}\mathbf{x}^{[s]} - \mathbf{b}\|_2
\end{aligned}$$

where $\flat$ follows from the facts that $\|\mathbf{g}^{[s]}\|_2 \leqslant \|g^{[s]}\|_2$ and $\|\vec{\mathbf{1}}\|_2 = \sqrt{K}$, and \$ from (**??**) and sub-multiplicativity of matrix norms. This concludes the derivation of (**??**).

Proof. [Proposition **??**] Let $\hat{g}^{[t]}$ be the approximated gradient of our scheme at iteration t. Since we are considering linear regression, it follows that

$$
\begin{aligned}
\left\|g^{[t]} - \hat{g}^{[t]}\right\|_2 &= 2\|\mathbf{A}^T(\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b}) - \mathbf{A}^T(\mathbf{S}_{\mathbf{\Pi}}^T \mathbf{S}_{\mathbf{\Pi}})(\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b})\|_2 \\
&= 2\|\mathbf{A}^T(\mathbf{I}_N - \mathbf{S}_{\mathbf{\Pi}}^T \mathbf{S}_{\mathbf{\Pi}})(\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b})\|_2 \\
&\leqslant 2\|\mathbf{A}\|_2 \cdot \|\mathbf{I}_N - \mathbf{S}_{\mathbf{\Pi}}^T \mathbf{S}_{\mathbf{\Pi}}\|_2 \cdot \|\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b}\|_2 \\
&= 2\|\mathbf{A}\|_2 \cdot \|\mathbf{U}^T(\mathbf{I}_N - \mathbf{S}_{\mathbf{\Pi}}^T \mathbf{S}_{\mathbf{\Pi}})\mathbf{U}\|_2 \cdot \|\mathbf{A}\mathbf{x}^{[s]} - \mathbf{b}\|_2 \\
&= 2\|\mathbf{A}\|_2 \cdot \|\mathbf{I}_d - \mathbf{U}^T \mathbf{S}_{\mathbf{\Pi}}^T \mathbf{S}_{\mathbf{\Pi}} \mathbf{U}\|_2 \cdot \|\mathbf{A}\mathbf{x}^{[s]} - \mathbf{b}\|_2 \\
&\overset{\flat}{\leqslant} 2\epsilon \cdot \|\mathbf{A}\|_2 \cdot \|\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b}\|_2
\end{aligned}
$$

where in $\flat$ we invoked the fact that $\mathbf{S}_{\mathbf{\Pi}}$ satisfies (**??**). Our approximate GC approach therefore (w.h.p.) satisfies (**??**), with $\mathrm{err}(\mathbf{G}_{(\mathcal{I})}) = \epsilon/\sqrt{K}$ $\qquad\square$

APPENDIX B
PROOFS OF SECTION **??**

In this appendix, we present two lemmas which we use to bound the entries of $\hat{\mathbf{V}} := \hat{\mathbf{H}}\mathbf{D}\mathbf{U}$, and its *leverage scores* $\ell_i := \|\hat{\mathbf{V}}_{(i)}\|_2^2$, for which $\sum_{i=1}^N \ell_i = d$. Leverage scores induce a sampling distribution which has proven to be useful in linear regression [**?**], [**?**], [**?**], [**?**] and GC [**?**]. From these lemmas, we deduce that the leverage scores of $\hat{\mathbf{H}}\mathbf{D}\mathbf{A}$ are close to being uniform, implying that the *block leverage scores* [**?**], [**?**] are also uniform, which is precisely what Lemma **??** states.

Lemma **??** is a variant of the Flattening Lemma [**?**], [**?**], a key result to Hadamard based sketching algorithms, which justifies uniform sampling. In the proof, we make use of the Azuma-Hoeffding inequality; a concentration result for the values of martingales that have bounded differences. We also recall a matrix Chernoff bound, which we apply to prove our ℓ_2-s.e. guarantees. Finally, we present proofs of Proposition **??** and Theorems **??**, **??**.

Lemma 7 (Azuma-Hoeffding Inequality, [**?**]). *For zero mean random variable Z_i (or $Z_0, Z_1, \cdots, Z_m$ a martingale sequence of random variables), bounded above by $|Z_i| \leqslant \beta_i$ for all i with probability 1, we have*

$$
\Pr\left[\left|\sum_{j=0}^m Z_j\right| > t\right] \leqslant 2\exp\left\{\frac{-t^2}{2 \cdot \left(\sum_{j=0}^m \beta_j^2\right)}\right\}.
$$

Theorem 8 (Matrix Chernoff Bound, [**?**, Fact 1]). *Let $\mathbf{X}_1, \cdots, \mathbf{X}_q$ be independent copies of a symmetric random matrix $\mathbf{X} \in \mathbb{R}^{d \times d}$, with $\mathbb{E}[\mathbf{X}] = 0, \|\mathbf{X}\|_2 \leqslant \gamma, \|\mathbb{E}[\mathbf{X}^T\mathbf{X}]\|_2 \leqslant \sigma^2$. Let $\mathbf{Z} = \frac{1}{q}\sum_{i=1}^q \mathbf{X}_i$. Then, $\forall \epsilon > 0$:*

$$
\Pr\left[\|\mathbf{Z}\|_2 > \epsilon\right] \leqslant 2d \cdot \exp\left(-\frac{q\epsilon^2}{\sigma^2 + \gamma\epsilon/3}\right). \tag{24}
$$

Lemma 8 (Flattening Lemma). *For $\mathbf{y} \in \mathbb{R}^N$ a fixed (orthonormal) column vector of $\mathbf{U}$, and $\mathbf{D} \in \{0, \pm 1\}^{N \times N}$ with random equi-probable diagonal entries of ± 1, we have:*

$$
\Pr\left[\|\hat{\mathbf{H}}\mathbf{D} \cdot \mathbf{y}\|_\infty > C\sqrt{\log(Nd/\delta)/N}\right] \leqslant \frac{\delta}{2d} \tag{25}
$$

for $0 < C \leqslant \sqrt{2 + \log(16)/\log(Nd/\delta)}$ a constant.

Proof. [Lemma **??**] Fix i and define $Z_j = \hat{\mathbf{H}}_{ij}\mathbf{D}_{jj}\mathbf{y}_j$ for each $j \in \mathbb{N}_N$, which are independent random variables. Since $\mathbf{D}_{jj} = \vec{D}_j$ are i.i.d. entries with zero mean, so are Z_j. Furthermore $|Z_j| \leqslant |\hat{\mathbf{H}}_{ij}| \cdot |\mathbf{D}_{jj}| \cdot |\mathbf{y}_j| = \frac{|\mathbf{y}_j|}{\sqrt{N}}$, and note that

$$
\sum_{j=1}^N Z_j = (\hat{\mathbf{H}}\mathbf{D}\mathbf{y})_i = \sum_{j=1}^N \hat{\mathbf{H}}_{ij}\mathbf{D}_{jj}\mathbf{y}_j = \langle\hat{\mathbf{H}}_{(i)} \odot \overbrace{\mathrm{diag}(\mathbf{D})}^{\vec{D}}, \mathbf{y}\rangle
$$

where $\odot$ is the Hadamard product. By Lemma **??**

$$
\begin{aligned}
\Pr\left[\left|\sum_{j=1}^N Z_j\right| > \rho\right] &\leqslant 2\exp\left\{\frac{-\rho^2/2}{\sum_{j=1}^N (\mathbf{y}_j/\sqrt{N})^2}\right\} \\
&= 2\exp\left\{\frac{-N\rho^2}{2 \cdot \langle\mathbf{y}, \mathbf{y}\rangle}\right\} \overset{\flat}{=} 2 \cdot e^{-N\rho^2/2}
\end{aligned} \tag{26}
$$

where $\flat$ follows from the fact that $\mathbf{y}$ is a column of $\mathbf{U}$. By setting $\rho = C\sqrt{\frac{\log(Nd/\delta)}{N}}$, we get

$$\Pr\left[\Big|\sum_{j=1}^{N} Z_j\Big| > C\sqrt{\frac{\log(Nd/\delta)}{N}}\right] \leqslant 2\exp\left\{-\frac{C^2\log(Nd/\delta)}{2}\right\}$$

$$= 2\left(\frac{\delta}{Nd}\right)^{C^2/2} \overset{\natural}{\leqslant} \frac{\delta}{2Nd}$$

where $\natural$ follows from the upper bound on C. By applying the union bound over all $i \in \mathbb{N}_N$, we attain (**??**). $\qquad\square$

Lemma 9. *For all $i \in \mathbb{N}_N$ and $\{\mathbf{e}_i\}_{i=1}^N$ the standard basis:*

$$\Pr\left[\sqrt{\ell_i} \leqslant C\sqrt{d\log(Nd/\delta)/N}\right] \geqslant 1 - \delta/2$$

for $\ell_i = \|\hat{\mathbf{V}}_{(i)}\|_2^2$ the ith leverage score of $\hat{\mathbf{V}} = \hat{\mathbf{H}}\mathbf{D}\mathbf{U}$.

Proof. [Lemma **??**] It is straightforward that the columns of $\hat{\mathbf{V}}$ form an orthonormal basis of $\mathbf{A}$, thus Lemma **??** implies that for $j \in \mathbb{N}_d$

$$\Pr\left[\|\hat{\mathbf{V}} \cdot \mathbf{e}_j\|_\infty > C\sqrt{\log(Nd/\delta)/N}\right] \leqslant \frac{\delta}{2d} \; .$$

By applying the union bound over all entries of $\hat{\mathbf{V}}^{(j)} = \hat{\mathbf{V}} \cdot \mathbf{e}_j$

$$\Pr\left[\overbrace{|\mathbf{e}_i^T \cdot \hat{\mathbf{V}} \cdot \mathbf{e}_j|}^{|(\hat{\mathbf{H}}\mathbf{D}\mathbf{U})_{ij}|} > C\sqrt{\frac{\log(Nd/\delta)}{N}}\right] \leqslant d \cdot \frac{\delta}{2d} = \delta/2 \; . \tag{27}$$

We manipulate the argument of the above bound to obtain

$$\|\mathbf{e}_i^T \cdot \hat{\mathbf{V}}\|_2 = \left(\sum_{j=1}^{d}(\hat{\mathbf{H}}\mathbf{D}\mathbf{U})_{ij}^2\right)^{1/2} > C\sqrt{d \cdot \frac{\log(Nd/\delta)}{N}} \; ,$$

which can be viewed as a scaling of the random variable entries of $\hat{\mathbf{V}}$. The probability of the complementary event is therefore

$$\Pr\left[\|\mathbf{e}_i^T \cdot \hat{\mathbf{V}}\|_2 \leqslant C\sqrt{d\log(Nd/\delta)/N}\right] \geqslant 1 - \delta/2$$

and the proof is complete. $\qquad\square$

Remark 1. *The complementary probable event of (**??**) can be interpreted as 'every entry of $\hat{\mathbf{V}}$ is small in absolute value'.*

Proof. [Lemma **??**] For $\alpha := \eta d \cdot \log(Nd/\delta)/N$

$$\Pr\left[\tilde{\ell}_\iota \leqslant \tau \cdot \alpha\right] > \Pr\left[\{\ell_j \leqslant \alpha : \forall j \in \mathcal{K}_\iota\}\right] \overset{\Diamond}{>} (1 - \delta/2)^\tau$$

where $\eta = C^2$ and $\Diamond$ follows from Lemma **??**. By the binomial approximation, we have $(1 - \delta/2)^\tau \approx 1 - \tau\delta/2$. $\qquad\square$

Define the symmetric matrices

$$\mathbf{X}_i = \left(\mathbf{I}_d - \frac{N}{\tau} \cdot \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)}\right) = \left(\mathbf{I}_d - K \cdot \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)}\right) \tag{28}$$

where $\hat{\mathbf{V}}_{(\mathcal{K}^i)} = \hat{\mathbf{V}}_{(\mathcal{K}_\iota)}$ is the submatrix of $\hat{\mathbf{V}}$ corresponding to the ith sampling trial of our algorithm. Let $\mathbf{X}$ be the matrix r.v. of which the $\mathbf{X}_i$'s are independent copies. Note that the realizations $\mathbf{X}_i$ of $\mathbf{X}$ correspond to the sampling blocks of the event in (**??**).

To apply Theorem **??**, we show that the $\mathbf{X}_i$'s have zero mean, and we bound their ℓ_2-norm and variance. Their ℓ_2-norms are upper bounded by

$$
\begin{aligned}
\|\mathbf{X}_i\|_2 &\leqslant \|\mathbf{I}_d\|_2 + \|\frac{N}{\tau} \cdot \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)}\|_2 \\
&= 1 + \frac{N}{\tau} \cdot \|\hat{\mathbf{V}}_{(\mathcal{K}_\iota)}\|_2^2 \\
&\leqslant 1 + \frac{N}{\tau} \cdot \max_{\iota \in \mathbb{N}_K} \left\{ \|\mathbf{I}_{(\mathcal{K}_\iota)} \cdot \hat{\mathbf{V}}\|_2^2 \right\} \\
&\leqslant 1 + \frac{N}{\tau} \cdot \max_{\iota \in \mathbb{N}_K} \left\{ \|\mathbf{I}_{(\mathcal{K}_\iota)} \cdot \hat{\mathbf{V}}\|_F^2 \right\} \qquad\qquad\qquad\qquad (29) \\
&\overset{\$}{\leqslant} 1 + \frac{N}{\tau} \cdot \left(|\mathcal{K}_\iota| \cdot \max_{j \in \mathbb{N}_N} \left\{ \|\mathbf{e}_j^T \cdot \hat{\mathbf{V}}\|_2^2 \right\} \right) \\
&\leqslant 1 + \frac{N}{\tau} \cdot \left(\tau \cdot (\eta \cdot d \log(Nd/\delta)/N) \right) \qquad\qquad\quad [\text{Lemma } \textbf{??}] \\
&= 1 + \eta \cdot d \log(Nd/\delta) \qquad\qquad\qquad\qquad\qquad\qquad (30) \\
&= 1 + N\alpha
\end{aligned}
$$

for $\alpha = \eta d \cdot \log(Nd/\delta)/N$ where in $\$$ we used the fact that

$$
\|\mathbf{I}_{(\mathcal{K}_\iota)} \cdot \hat{\mathbf{V}}\|_F^2 = \sum_{j \in \mathcal{K}_\iota} \|\mathbf{e}_j^T \cdot \hat{\mathbf{V}}\|_2^2 \leqslant |\mathcal{K}_\iota| \cdot \max_{j \in \mathcal{K}_\iota} \left\{ \|\mathbf{e}_j^T \cdot \hat{\mathbf{V}}\|_2^2 \right\} .
$$

From the above derivation, it follows that

$$
\begin{aligned}
\|\hat{\mathbf{V}}_{(\mathcal{K}^i)}\|_2^2 &= \|\hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)}\|_2 \\
&\leqslant \frac{\tau}{N} \cdot (1 + \eta \cdot d \log(Nd/\delta) - \|\mathbf{I}_d\|_2) \\
&= \tau \eta d / N \cdot \log(Nd/\delta) \\
&= \tau \alpha
\end{aligned}
$$

for all $\iota \in \mathbb{N}_K$. By setting $\tau = 1$, we get an upper bound on the squared ℓ_2-norm of the rows of $\hat{\mathbf{V}}$:

$$
\|\hat{\mathbf{V}}_l\|_2^2 = \|\hat{\mathbf{V}}_l \hat{\mathbf{V}}_l^T\|_2 = \|\hat{\mathbf{V}}_l^T \hat{\mathbf{V}}_l\|_2 \leqslant \alpha \qquad\qquad\qquad (31)
$$

where $\hat{\mathbf{V}}_l = \hat{\mathbf{V}}_{(l)}$, for all $l \in \mathbb{N}_N$.

Next, we compute $\mathbf{E} := \mathbb{E}[\mathbf{X}^T\mathbf{X} + \mathbf{I}_d]$ and its eigenvalues. By the definition of $\mathbf{X}$ and its realizations:

$$
\begin{aligned}
\mathbf{X}_i^T \mathbf{X}_i &= \left(\mathbf{I}_d - N/\tau \cdot \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)} \right)^T \cdot \left(\mathbf{I}_d - N/\tau \cdot \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)} \right) \\
&= \mathbf{I}_d - 2 \cdot \frac{N}{\tau} \cdot \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)} + \left(\frac{N}{\tau} \right)^2 \cdot \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)} \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)}
\end{aligned}
$$

thus $\mathbf{E}$ is evaluated as follows:

$$
\begin{aligned}
\mathbb{E}[\mathbf{X}^T\mathbf{X} + \mathbf{I}_d] &= 2\mathbf{I}_d - 2 \cdot (N/\tau) \cdot \mathbb{E}\left[\hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)} \right] + (N/\tau)^2 \cdot \mathbb{E}\left[\hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)} \hat{\mathbf{V}}_{(\mathcal{K}^i)}^T \hat{\mathbf{V}}_{(\mathcal{K}^i)} \right] \\
&= 2\mathbf{I}_d - 2 \cdot (N/\tau) \cdot \left(\sum_{j=1}^K K^{-1} \cdot \hat{\mathbf{V}}_{(\mathcal{K}_j)}^T \hat{\mathbf{V}}_{(\mathcal{K}_j)} \right) + (N/\tau)^2 \cdot \left(\sum_{j=1}^K K^{-1} \cdot \hat{\mathbf{V}}_{(\mathcal{K}_j)}^T \left(\hat{\mathbf{V}}_{(\mathcal{K}_j)} \hat{\mathbf{V}}_{(\mathcal{K}_j)}^T \right) \hat{\mathbf{V}}_{(\mathcal{K}_j)} \right) \\
&= 2\mathbf{I}_d - 2 \cdot \left(\sum_{l=1}^N \hat{\mathbf{V}}_l^T \hat{\mathbf{V}}_l \right) + (N/\tau) \cdot \left(\sum_{l=1}^N \hat{\mathbf{V}}_l^T \left(\hat{\mathbf{V}}_l \hat{\mathbf{V}}_l^T \right) \hat{\mathbf{V}}_l \right) \\
&= K \cdot \left(\sum_{l=1}^N \langle \hat{\mathbf{V}}_l, \hat{\mathbf{V}}_l \rangle \cdot \hat{\mathbf{V}}_l^T \hat{\mathbf{V}}_l \right)
\end{aligned}
$$

where in the last equality we invoked $\sum_{l=1}^N \hat{\mathbf{V}}_l^T \hat{\mathbf{V}}_l = \mathbf{I}_d$.

In order to bound the variance of the matrix random variable $\mathbf{X}$, we bound the largest eigenvalue of $\mathbf{E}$; by comparing it to the matrix

$$
\mathbf{F} = K\alpha \cdot \left(\sum_{l=1}^N \hat{\mathbf{V}}_l^T \hat{\mathbf{V}}_l \right) = K\alpha \cdot \mathbf{I}_d
$$

whose eigenvalue $K\alpha$ is of algebraic multiplicity d. It is clear that $\mathbf{E}$ and $\mathbf{F}$ are both real and symmetric; thus they admit an eigendecomposition of the form $\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T$. Note also that for all $\mathbf{y} \in \mathbb{R}^d$:

$$
\begin{aligned}
\mathbf{y}^T\mathbf{E}\mathbf{y} &= K \cdot \mathbf{y}^T \left(\sum_{l=1}^{N} \hat{\mathbf{V}}_l^T \left(\hat{\mathbf{V}}_l \hat{\mathbf{V}}_l^T \right) \hat{\mathbf{V}}_l \right) \mathbf{y} \\
&\overset{\sharp}{=} K \cdot \sum_{l=1}^{N} \langle \mathbf{y}, \hat{\mathbf{V}}_l \rangle^2 \cdot \|\hat{\mathbf{V}}_l\|_2^2 \\
&\overset{\flat}{\leqslant} K\alpha \cdot \sum_{l=1}^{N} \langle \mathbf{y}, \hat{\mathbf{V}}_l \rangle^2 \\
&= K\alpha \cdot \sum_{l=1}^{N} \mathbf{y}^T \hat{\mathbf{V}}_l^T \cdot \hat{\mathbf{V}}_l \mathbf{y} \\
&= \mathbf{y}^T \left(K\alpha \cdot \sum_{l=1}^{N} \hat{\mathbf{V}}_l^T \cdot \hat{\mathbf{V}}_l \right) \mathbf{y} \\
&= \mathbf{y}^T \mathbf{F} \mathbf{y}
\end{aligned}
\tag{33}
$$

where in $\flat$ we invoked (**??**). By $\sharp$ we conclude that $\mathbf{y}^T\mathbf{E}\mathbf{y} \geqslant 0$, thus $\mathbf{F} \succeq \mathbf{E} \succeq 0$.

Let $\mathbf{w}_i, \mathbf{z}_i$ be the unit-norm eigenvectors of $\mathbf{E}, \mathbf{F}$ corresponding to their respective i^{th} largest eigenvalue. Then

$$
\mathbf{w}_i^T \left(\mathbf{Q_E}\boldsymbol{\Lambda_E}\mathbf{Q_E}^T \right) \mathbf{w}_i = \mathbf{e}_i^T \cdot \boldsymbol{\Lambda_E} \cdot \mathbf{e}_i = \lambda_i
$$

and by (**??**) we bound this as follows:

$$
\lambda_i = \mathbf{w}_i^T \mathbf{E} \mathbf{w}_i \leqslant K\alpha \cdot \sum_{l=1}^{N} \langle \mathbf{w}_i, \hat{\mathbf{V}}_l \rangle^2 \, .
$$

Since

$$
\mathbf{w}_1 = \arg \max_{\substack{\mathbf{v} \in \mathbb{R}^d \\ \|\mathbf{v}\|_2 = 1}} \left\{ \mathbf{v}^T \mathbf{E} \mathbf{v} \right\} \implies \|\mathbf{E}\|_2 = \lambda_1 = \mathbf{w}_1^T \mathbf{E} \mathbf{w}_1 \, ,
$$

and $\mathbf{F} \succeq \mathbf{E} \geqslant 0$, it follows that

$$
\|\mathbf{E}\|_2 = \mathbf{w}_1^T \mathbf{E} \mathbf{w}_1 \leqslant \mathbf{w}_1^T \mathbf{F} \mathbf{w}_1 \leqslant \arg \max_{\substack{\mathbf{v} \in \mathbb{R}^d \\ \|\mathbf{v}\|_2 = 1}} \left\{ \mathbf{v}^T \mathbf{F} \mathbf{v} \right\} = \|\mathbf{F}\|_2 = K\alpha \, .
$$

In turn, this gives us

$$
\begin{aligned}
\|\mathbb{E}[\mathbf{X}^T\mathbf{X}]\|_2 &= \|\mathbf{E} - \mathbf{I}_d\|_2 \\
&\leqslant \|\mathbf{E}\|_2 + \|\mathbf{I}_d\|_2 \\
&\leqslant \|\mathbf{F}\|_2 + 1 \\
&= K\alpha + 1 \\
&\leqslant \eta K \frac{d}{N} \log(Nd/\delta) + 1 \\
&= \eta \frac{d}{\tau} \log(Nd/\delta) + 1
\end{aligned}
\tag{34}
$$

hence $\|\mathbb{E}[\mathbf{X}^T\mathbf{X}]\|_2 = O\big(\frac{d}{\tau} \log(Nd/\delta)\big)$.

We now have everything we need to apply Theorem **??**.

Proposition 4. *The block-SRHT $\mathbf{S}_{\hat{\boldsymbol{\Pi}}}$ guarantees*

$$
\Pr\left[\|\mathbf{I}_d - \mathbf{U}^T\mathbf{S}_{\hat{\boldsymbol{\Pi}}}^T\mathbf{S}_{\hat{\boldsymbol{\Pi}}}\mathbf{U}\|_2 > \epsilon \right] \leqslant 2d \cdot \exp\left\{ \frac{-\epsilon^2 \cdot q}{\Theta\left(\frac{d}{\tau} \cdot \log(Nd/\delta)\right)} \right\}
$$

for any $\epsilon > 0$, and $q = r/\tau > d/\tau$.

Proof. [Proposition **??**] Let $\{\mathbf{X}_i\}_{i=1}^q$ as defined in (**??**) denote q block samples. Let $j(i)$ denote the index of the submatrix which was sampled at the ith random trial, *i.e.* $\mathcal{K}_{j(i)} = \mathcal{K}_{j(i)}^i$. Then

$$
\begin{aligned}
\mathbf{Z} &= \frac{1}{q} \sum_{i=1}^t \mathbf{X}_{j(i)} \\
&= \frac{1}{q} \cdot \sum_{i=1}^q \left(\mathbf{I}_d - \frac{N}{\tau} \cdot \hat{\mathbf{V}}_{(\mathcal{K}_{j(i)})}^T \hat{\mathbf{V}}_{(\mathcal{K}_{j(i)})} \right) \\
&= \mathbf{I}_d - \sum_{i=1}^q \left(\sqrt{N/r} \cdot \hat{\mathbf{V}}_{(\mathcal{K}_{j(i)})} \right)^T \cdot \left(\sqrt{N/r} \cdot \hat{\mathbf{V}}_{(\mathcal{K}_{j(i)})} \right) \\
&= \mathbf{I}_d - \sum_{i=1}^q \left(\sqrt{N/r} \cdot \mathbf{I}_{(\mathcal{K}_{j(i)})} \cdot \hat{\mathbf{V}} \right)^T \cdot \left(\sqrt{N/r} \cdot \mathbf{I}_{(\mathcal{K}_{j(i)})} \cdot \hat{\mathbf{V}} \right) \\
&= \mathbf{I}_d - \left(\widetilde{\boldsymbol{\Omega}} \hat{\mathbf{H}} \mathbf{D} \mathbf{U} \right)^T \cdot \left(\widetilde{\boldsymbol{\Omega}} \hat{\mathbf{H}} \mathbf{D} \mathbf{U} \right) \\
&= \mathbf{I}_d - \mathbf{U}^T \mathbf{S}_{\hat{\boldsymbol{\Pi}}}^T \mathbf{S}_{\hat{\boldsymbol{\Pi}}} \mathbf{U} \,.
\end{aligned}
$$

We apply Lemma **??** by fixing the terms we bounded: (**??**) $\gamma = \eta d \log(Nd/\delta) + 1$, (**??**) $\sigma^2 = \eta \frac{d}{\tau} \log(Nd/\delta) + 1$, and fix q and ϵ. The denominator of the exponent in (**??**) is then

$$
\begin{aligned}
\left(\eta d/\tau \cdot \log(Nd/\delta) + 1 \right) + \left((\eta d \log(Nd/\delta) + 1) \cdot \epsilon/3 \right) &= \\
&= \eta d/\tau \cdot \log(Nd/\delta) \cdot \left(1 + \epsilon\tau/3 \right) + (1 + \epsilon/3) \\
&= \Theta\left(\frac{d}{\tau} \log(Nd/\delta) \right)
\end{aligned}
$$

and the proof is complete. $\qquad\square$

Proof. [Theorem **??**] By substituting q in the bound of Proposition **??** and taking the complementary event, we attain the statement.
$\qquad\square$

A. The Hadamard Transform

Remark 2. *The Hadamard matrix is a real analog of the discrete Fourier matrix, and there exist matrix multiplications algorithms for the Hadamard transform which resemble the FFT algorithm. Recall that the Fourier matrix represents the characters of the cyclic group of order N. In this case, $\hat{\mathbf{H}}_N$ represents the characters of the group $(\mathbb{Z}_2^N, +)$, where $\mathbb{Z}_2^n \cong \mathbb{Z}_N$. For both of these transforms, it is precisely through this algebraic structure which one can achieve a matrix-vector multiplication in $\mathcal{O}(N \log N)$ arithmetic operations.*

Recall that the characters of a group G, form an orthonormal basis of the vector space of functions over the Boolean hypercube, *i.e.* $\mathcal{F}_n = \{f : \{0,1\}^n \to \mathbb{R}\}$, and it is the Fourier basis. Furthermore, when working over groups of characteristic 2, *e.g.* $\mathbb{F}_{2^q} \cong \mathbb{F}_2^q$ for $q \in \mathbb{Z}_+$, we can move everything so that the underlying field is $\mathbb{R}$. Specifically, we map the elements of the binary field to $\mathbb{R}$ by applying $f(y) = 1 - 2y$. This gives us $f : \{0,1\} \mapsto \{+1, -1\} \subseteq \mathbb{R}$, and we can work with addition and multiplication over $\mathbb{R}$.

We note that there is a bijective correspondence between the characters of $\mathbb{Z}_m$ and the mth root of unity, which is precisely how we get an orthonormal (Fourier) basis. In the case where m is not a power of two, we have a basis with complex elements, which violates (c) in the list of properties we seek to satisfy. This is why the Hadamard matrix is appropriate for our application, and why we do not consider a general discrete Fourier transform.

B. Recursive Kronecker Product Orthogonal Matrices

In this subsection, we show that multiplying a vector of length N with $\boldsymbol{\Pi} = \boldsymbol{\Pi}_k^{\otimes \lceil \log_k(N) \rceil}$ for $\boldsymbol{\Pi}_k \in O_k(\mathbb{R})$ and $k \in \mathbb{Z}_{>2}$, takes $\mathcal{O}(Nk^2 \log_k N)$ elementary operations. Therefore, multiplying $\mathbf{A} \in \mathbb{R}^{N \times d}$ with $\boldsymbol{\Pi}$ takes $\mathcal{O}(Ndk^2 \log_k N)$ operations. We follow a similar analysis to that of [**?**, Section 6.10.2].

For $C(N)$ the number of elementary operations involved in carrying out the above matrix-vector multiplication, the basic recursion relationship is

$$C(N) = k(C/k) + NC(1) \tag{35}$$

where $C(1) = \zeta k^2$, for $\zeta > 0$ a constant.

For $p = \lceil \log_k(N) \rceil$, we have the following relationship:

$$T(p) = \frac{C(N)}{N} \quad \implies \quad C(N) = NT(p). \tag{36}$$

Then, $p - 1 = \lceil \log_k(N/k) \rceil$, which gives us

$$T(p-1) = \frac{C(N/k)}{N/k} = k\frac{C(N/k)}{N} \quad \implies \quad NT(p-1) = kC(N/k). \tag{37}$$

By substituting (**??**) into (**??**), we get

$$C(N) = NT(p) = NT(p-1) + NC(1),$$

thus $T(p) = T(p-1) + C(1)$, which implies that $T(p)$ is linear. Therefore $T(p) = pC(1) = \zeta k^2 p$, and from (**??**) we conclude that the total number of elementary operations is

$$C(N) = NT(p) = N\zeta k^2 p = N\zeta k^2 \lceil \log_k(N) \rceil = \mathcal{O}(Nk^2 \log_k N).$$

APPENDIX C
PROOFS OF SECTION **??**

In this appendix, we present the proofs of Proposition **??** and Corollary **??**.

Proof. [Proposition **??**] Note that the optimization problem (**??**) is equivalent to

$$\xi_t^\star = \arg\min_{\xi \in \mathbb{R}} \left\{ \|\mathbf{A}\mathbf{x}^{[t+1]} - \mathbf{b}\|_2^2 \right\}. \tag{38}$$

If we cannot decrease further, the optimal solution to (**??**) will be 0, and we can never have $\xi_t < 0$, as this would imply that

$$\|\mathbf{A}\mathbf{x}^{[t+1]} - \mathbf{b}\|_2^2 = \|\mathbf{A}(\mathbf{x}^{[t]} - \xi_t \cdot g^{[t]})\mathbf{b}\|_2^2 > \|\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b}\|_2^2$$

which contradicts the fact that we are minimizing the objective function of (**??**). Specifically, if $\xi_t < 0$, we get an ascent step in (**??**), and a step-size $\xi_t = 0$ achieves a lower value. It therefore suffices to prove the given statement by solving (**??**).

We will first derive (**??**) for $L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$, and then show it is the same for the optimization problems $L_{\mathbf{\Pi}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$ and $L_{\mathbf{G}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$.

Recall that $\mathbf{x}^{[t+1]} \leftarrow \mathbf{x}^{[t]} - \xi_t \cdot g_{ls}^{[t]}$ for the least squares objective $L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$. From here onward, we denote the gradient update of the underlying objective function by g_t. We then reformulate the objective function of (**??**) as follows

$$\Delta_{t+1} := \|\mathbf{A}\mathbf{x}^{[t+1]} - \mathbf{b}\|_2^2 = \|\mathbf{A}(\mathbf{x}^{[t]} - \xi \cdot g^{[t]}) - \mathbf{b}\|_2^2.$$

By expanding the above expression, we get

$$\Delta_{t+1} = \xi^2 \cdot \|\mathbf{A}g_t\|_2^2 - 2\xi \cdot \left(g_t^T \mathbf{A}^T \mathbf{A}\mathbf{x}^{[t]} - g_t^T \mathbf{A}^T \mathbf{b} \right) + \\ + \left(\|\mathbf{A}\mathbf{x}^{[t]}\|_2^2 - 2\langle \mathbf{A}\mathbf{x}^{[t]}, \mathbf{b} \rangle + \langle \mathbf{b}, \mathbf{b} \rangle \right)$$

and by setting $\frac{\partial \Delta_{t+1}}{\partial \xi} = 0$ and solving for ξ, it follows that

$$\frac{\partial \Delta_{t+1}}{\partial \xi} = 2\xi \cdot \left(g_t^T \mathbf{A}^T \mathbf{A} g_t \right) - 2 \cdot \left(g_t^T \mathbf{A}^T \right) \left(\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b} \right) = 0$$

$$\implies \quad \xi_t^\star = \frac{\langle \mathbf{A}g_t, \mathbf{A}\mathbf{x}^{[t]} - \mathbf{b} \rangle}{\|\mathbf{A}g_t\|_2^2}, \tag{39}$$

which is the updated step-size we use at the next iteration. Since $\partial^2 \Delta_{t+1}/\partial\xi^2 = 2\|\mathbf{A}g_t\|_2^2 \geqslant 0$, we know that Δ_{t+1} is convex. Therefore, $\xi_t^\star$ derived in (**??**) is indeed the minimizer of Δ_{t+1}.

Now consider SD with the objective function $L_{\mathbf{\Pi}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$. The only thing that changes in the derivation, is that now we have $(\grave{\mathbf{A}} = \mathbf{\Pi}\mathbf{A}, \grave{\mathbf{b}} = \mathbf{\Pi}\mathbf{b})$ instead of $(\mathbf{A}, \mathbf{b})$. By replacing $(\grave{\mathbf{A}}, \grave{\mathbf{b}}) \leftarrow (\mathbf{A}, \mathbf{b})$ in (**??**), it follows that

$$\frac{\langle \grave{\mathbf{A}}g_t, \grave{\mathbf{A}}\mathbf{x}^{[t]} - \grave{\mathbf{b}} \rangle}{\|\grave{\mathbf{A}}g_t\|_2^2} = \frac{\langle \mathbf{A}g_t, \mathbf{A}\mathbf{x}^{[t]} - \mathbf{b} \rangle}{\|\mathbf{A}g_t\|_2^2} \tag{40}$$

as $\grave{\mathbf{A}}^T \grave{\mathbf{A}} = \mathbf{A}^T \mathbf{A}$ and $\grave{\mathbf{A}}^T \grave{\mathbf{b}} = \mathbf{A}^T \mathbf{b}$, since $\mathbf{\Pi} \in O_N(\mathbb{R})$. The step-sizes for the corresponding iterations are therefore identical.

Moreover, the only difference between the objective functions $L_{\mathbf{\Pi}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$ and $L_{\mathbf{G}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$ is the factor of $\sqrt{N/r}$. Let $\tilde{\mathbf{A}} = \mathbf{G}\mathbf{A}$ and $\tilde{\mathbf{b}} = \mathbf{G}\mathbf{b}$. Therefore, the step-size at iteration $t+1$ when considering the objective function $L_{\mathbf{G}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$ is

$$\frac{\langle \tilde{\mathbf{A}}g_t, \tilde{\mathbf{A}}\mathbf{x}^{[t]} - \tilde{\mathbf{b}} \rangle}{\|\tilde{\mathbf{A}}g_t\|_2^2} = \frac{N/r}{N/r} \cdot \frac{\langle \grave{\mathbf{A}}g_t, \grave{\mathbf{A}}\mathbf{x}^{[t]} - \grave{\mathbf{b}} \rangle}{\|\grave{\mathbf{A}}g_t\|_2^2}$$

$$\overset{\diamond}{=} \frac{\langle \mathbf{A}g_t, \mathbf{A}\mathbf{x}^{[t]} - \mathbf{b} \rangle}{\|\mathbf{A}g_t\|_2^2}$$

where $\diamondsuit$ follows from (**??**). $\qquad\square$

Proof. [Corollary **??**] We want to show that $\xi_t^\star$ according to (**??**), is a solution to (**??**). We know that the only difference in the induced sketching matrices $\mathbf{S}_{\boldsymbol{\Pi}}^{[t]}$ at each iteration are the resulting index sets $\mathcal{S}^{[t]}$, and the corresponding sampling and rescaling matrices $\widetilde{\boldsymbol{\Omega}}_{[t]}$.

To prove the given statement, since $\mathbf{S}_{\boldsymbol{\Pi}}^{[t]} = \widetilde{\boldsymbol{\Omega}}_{[t]}\boldsymbol{\Pi}$; and by Proposition (**??**) $\xi_t^\star$ is a solution to

$$\arg\min_{\xi\in\mathbb{R}}\left\{\left\|\boldsymbol{\Pi}\left(\mathbf{A}\hat{\mathbf{x}}^{[t+1]} - \mathbf{b}\right)\right\|_2^2\right\},$$

it suffices to show that $\mathbb{E}\left[\widetilde{\boldsymbol{\Omega}}_{[t]}^T\widetilde{\boldsymbol{\Omega}}_{[t]}\right] = \mathbf{I}_N$. This was proven in Lemma **??**. Hence, the proof is complete. $\qquad\square$

APPENDIX D
PROOFS OF SECTION **??**

In this appendix, we present the proofs of Theorems **??** and **??**, and Corollary **??**. We also present a counterexample to perfect secrecy of the SRHT.

Proof. [Theorem **??**] Denote the application of $\boldsymbol{\Pi}$ to a matrix $\mathbf{M}$ by $\mathsf{Enc}_{\boldsymbol{\Pi}}(\mathbf{M}) = \boldsymbol{\Pi}\mathbf{M}$. We will prove secrecy of this scheme, which then implies that a subsampled version of the transformed information is also secure. Let $\grave{\mathbf{A}} = \mathsf{Enc}_{\boldsymbol{\Pi}}(\mathbf{A})$ and $\grave{\mathbf{b}} = \mathsf{Enc}_{\boldsymbol{\Pi}}(\mathbf{b})$.

The adversaries' goal is to reveal $\mathbf{A}$. To prove that $\mathsf{Enc}_{\boldsymbol{\Pi}}$ is a well-defined security scheme, we need to show that an adversary cannot learn recover $\mathbf{A}$; with only knowledge of $(\grave{\mathbf{A}}, \grave{\mathbf{b}})$.

For a contradiction, assume an adversary is able to recover $\mathbf{A}$ after only observing $(\grave{\mathbf{A}}, \grave{\mathbf{b}})$. This means that it was able to obtain $\boldsymbol{\Pi}^{-1}$, as the only way to recover $\mathbf{A}$ from $\grave{\mathbf{A}}$ is by inverting the transformation of $\boldsymbol{\Pi}$: $\mathbf{A} = \boldsymbol{\Pi}^{-1} \cdot \grave{\mathbf{A}}$. This contradicts the fact that only $(\grave{\mathbf{A}}, \grave{\mathbf{b}})$ were observed. Thus, $\mathsf{Enc}_{\boldsymbol{\Pi}}$ is a well-defined security scheme.

It remains to prove perfect secrecy according to Definition **??**. Observe that for any $\bar{\mathbf{U}} \in \mathcal{M}$ and $\bar{\mathbf{Q}} \in \mathcal{C}$

$$\Pr_{\boldsymbol{\Pi}\xleftarrow{U}\mathcal{K}}\left[\mathsf{Enc}_{\boldsymbol{\Pi}}(\bar{\mathbf{U}}) = \bar{\mathbf{Q}}\right] = \Pr_{\boldsymbol{\Pi}\xleftarrow{U}\mathcal{K}}\left[\boldsymbol{\Pi}\cdot\bar{\mathbf{U}} = \bar{\mathbf{Q}}\right] = \tag{41}$$

$$= \Pr_{\boldsymbol{\Pi}\xleftarrow{U}\mathcal{K}}\left[\boldsymbol{\Pi} = \bar{\mathbf{Q}}\cdot\bar{\mathbf{U}}^{-1}\right] \stackrel{\sharp}{=} \frac{1}{|\tilde{O}_{\mathbf{A}}|} = \frac{1}{|\mathcal{K}|} \tag{42}$$

where $\sharp$ follows from the fact that $\bar{\mathbf{Q}}\cdot\bar{\mathbf{U}}^{-1}$ is fixed. Hence, for any $\mathbf{U}_0, \mathbf{U}_1 \in \mathcal{M}$ and $\bar{\mathbf{Q}} \in \mathcal{C}$ we have

$$\Pr_{\boldsymbol{\Pi}\xleftarrow{U}\mathcal{K}}\left[\mathsf{Enc}_{\boldsymbol{\Pi}}(\mathbf{U}_0) = \bar{\mathbf{Q}}\right] = \frac{1}{|\mathcal{K}|} = \Pr_{\boldsymbol{\Pi}\xleftarrow{U}\mathcal{K}}\left[\mathsf{Enc}_{\boldsymbol{\Pi}}(\mathbf{U}_1) = \bar{\mathbf{Q}}\right]$$

as required by Definition **??**. This completes the proof. $\qquad\square$

We note that through the SVD of $\grave{\mathbf{A}}$, the adversaries can learn the singular values and right singular vectors of $\mathbf{A}$, since

$$\grave{\mathbf{A}} = (\boldsymbol{\Pi}\cdot\mathbf{U}_{\mathbf{A}})\cdot\boldsymbol{\Sigma}_{\mathbf{A}}\cdot\mathbf{V}_{\mathbf{A}}^T = \mathbf{U}_{\grave{\mathbf{A}}}\cdot\boldsymbol{\Sigma}_{\mathbf{A}}\cdot\mathbf{V}_{\mathbf{A}}^T . \tag{43}$$

Recall that the singular values are unique and, for distinct positive singular values, the corresponding left and right singular vectors are also unique up to a sign change of both columns. We assume w.l.o.g. that $\mathbf{V}_{\grave{\mathbf{A}}} = \mathbf{V}_{\mathbf{A}}$ and $\mathbf{U}_{\grave{\mathbf{A}}} = \boldsymbol{\Pi}\cdot\mathbf{U}_{\mathbf{A}}$.

Geometrically, the encoding $\mathsf{Enc}_{\boldsymbol{\Pi}}$ changes the orthonormal basis of $\mathbf{U}_{\mathbf{A}}$ to $\mathbf{U}_{\grave{\mathbf{A}}}$, by rotating it or reflecting it; when $\det(\boldsymbol{\Pi})$ is +1 or -1 respectively. Of course, there are infinitely many ways to do so, which is what we are relying the security of this approach on.

Furthermore, unless $\mathbf{U}_{\mathbf{A}}$ has some special structure (*e.g.*, triangular, symmetric, etc.), one cannot use an off-the-shelf factorization to reveal $\mathbf{U}_{\mathbf{A}}$. Even though a lot can be revealed about $\mathbf{A}$, *i.e.* $\boldsymbol{\Sigma}_{\mathbf{A}}$ and $\mathbf{V}_{\mathbf{A}}$, we showed that it is not possible to reveal $\mathbf{U}_{\mathbf{A}}$; hence nor $\mathbf{A}$, without knowledge of $\boldsymbol{\Pi}$.

Proof. [Corollary **??**] The proof is identical to that of Lemma **??**. The only difference is that the random variable entries $\tilde{Z}_j = \tilde{\mathbf{H}}_{ij}\mathbf{D}_{jj}\mathbf{y}_j$ for $j \in \mathbb{N}_N$ and the fixed i now differ, though they still meet the same upper bound

$$|\tilde{Z}_j| \leqslant |\tilde{\mathbf{H}}_{ij}| \cdot |\mathbf{D}_{jj}| \cdot |\mathbf{y}_j| = \frac{|\mathbf{y}_j|}{\sqrt{N}} .$$

Since (**??**) holds true, the guarantees implied by flattening lemma also do, thus the sketching properties of the SRHT are maintained. $\qquad\square$

Remark 3. *Since the Lemma **??** and Corollary **??** give the same result for the block-SRHT and garbled block-SRHT respectively, it follows that Theorem **??** also holds for the garbled block-SRHT.*

Proof. [Theorem **??**] Assume w.l.o.g. that a computationally bounded adversary observes $\tilde{\boldsymbol{\Pi}}\mathbf{A}$, for which $\tilde{\mathbf{A}}_r = \mathbf{S}_{\boldsymbol{\Pi}} \cdot \mathbf{A} = \tilde{\boldsymbol{\Omega}} \cdot (\tilde{\boldsymbol{\Pi}}\mathbf{A})$ is the resulting sketch of Algorithm **??**, for $\tilde{\boldsymbol{\Pi}} \in \tilde{H}_N$. To invert the transformation of $\tilde{\boldsymbol{\Pi}}$, the adversary needs knowledge

of the components of $\tilde{\Pi}$, *i.e.* $\hat{\mathbf{H}}$ and $\mathbf{P}$. Assume for a contradiction that there exists a probabilistic polynomial-time algorithm which, is able to recover $\mathbf{A}$ from $\tilde{\Pi}\mathbf{A}$. This means that it has revealed $\mathbf{P}$, so that it can compute

$$\overbrace{(\mathbf{D}\hat{\mathbf{H}}\mathbf{P}^T)}^{\tilde{\Pi}^T=\tilde{\Pi}^{-1}} \cdot (\mathbf{P}\hat{\mathbf{H}}\mathbf{D}) \cdot \mathbf{A} = \tilde{\Pi}^{-1} \cdot \tilde{\Pi} \cdot \mathbf{A} = \mathbf{A} \ ,$$

which contradicts the assumption that the permutation $\mathbf{P}$ is a s-PRP. Specifically, recovering $\mathbf{A}$ by observing $\tilde{\Pi}\mathbf{A}$ requires finding $\mathbf{P}$ in polynomial time. $\qquad\square$

Finally, we show that $\hat{g}^{[t]} = g_{ls}^{[t]}$, which we claimed in Subsection **??**. Since $\Pi \in O_N(\mathbb{R})$ for the suggested projections (except that random Rademacher projection), we have $\Pi^T\Pi = \mathbf{I}_N$. It then follows that

$$\begin{aligned}
\hat{g}^{[t]} &= 2\sum_{j=1}^{K} \tilde{\mathbf{A}}_j^T \left(\tilde{\mathbf{A}}_j \mathbf{x}^{[t]} - \tilde{\mathbf{b}}_j \right) \\
&= (\Pi\mathbf{A})^T \left(\Pi\mathbf{A}\mathbf{x}^{[t]} - \Pi\mathbf{b} \right) \\
&= \mathbf{A}^T \left(\Pi^T\Pi \right) \left(\mathbf{A}\mathbf{x}^{[t]} - \mathbf{b} \right) \\
&= g_{ls}^{[t]}
\end{aligned} \tag{44}$$

and this completes the derivation.

A. Counterexample to Perfect Secrecy of the SRHT

Here, we present an explicit example for the SRHT (which also applies to the block-SRHT), which contradicts Definition **??**. Therefore, the SRHT cannot provide perfect secrecy.

Consider the simple case where $N = 2$, and assume that $\hat{\mathbf{H}}_2 \in \tilde{O}_{\mathbf{A}}$. Since $(\tilde{O}_{\mathbf{A}}, \cdot)$ is a multiplicative subgroup of $\mathrm{GL}_2(\mathbb{R})$, we have $\mathbf{I}_2 \in \tilde{O}_{\mathbf{A}}$. Let $\mathbf{U}_0 = \mathbf{I}_2$ and $\mathbf{U}_1 = \hat{\mathbf{H}}_2$.

For d_1, d_2 i.i.d. Rademacher random variables and

$$\mathbf{D} = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix},$$

it follows that

$$\mathbf{C}_0 = \left(\hat{\mathbf{H}}_2\mathbf{D} \right) \cdot \mathbf{U}_0 = \hat{\mathbf{H}}_2\mathbf{D} = \frac{1}{2}\begin{pmatrix} d_1 & -d_2 \\ d_1 & d_2 \end{pmatrix}$$

and

$$\begin{aligned}
\mathbf{C}_1 = \left(\hat{\mathbf{H}}_2\mathbf{D} \right) \cdot \mathbf{U}_1 &= \frac{1}{2}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} d_1 & -d_1 \\ d_2 & d_2 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} d_1 - d_2 & -d_1 - d_2 \\ d_1 + d_2 & -d_1 + d_2 \end{pmatrix}.
\end{aligned}$$

It is clear that $\mathbf{C}_0$ always has precisely two distinct entries, while $\mathbf{C}_1$ has three distinct entries; with 0 appearing twice for any pair $d_1, d_2 \in \{\pm 1\}$. Therefore, depending on the observed transformed matrix, we can disregard one of $\mathbf{U}_0$ and $\mathbf{U}_1$ as being a potential choice for Π.

For instance, if $\bar{\mathbf{C}}$ is the observed matrix and it has two zero entries, then

$$\Pr_{\Pi \xleftarrow{U} H_2} \left[\Pi \cdot \mathbf{U}_1 = \bar{\mathbf{C}} \right] > \Pr_{\Pi \xleftarrow{U} H_2} \left[\Pi \cdot \mathbf{U}_0 = \bar{\mathbf{C}} \right] = 0$$

which contradicts (**??**).

Note that even if we apply a permutation, as in the case of the garbled block-SRHT, we still get the same conclusion. Hence, the garbled block-SRHT also does not achieve perfect secrecy.

B. Analogy with the One-Time Pad

It is worth noting that the encryption resulting by the multiplication with $\mathbf{\Pi}$; under the assumptions made in Theorem **??**, bares a strong resemblance with the one-time pad (OTP), which is the optimum cryptosystem with theoretically perfect secrecy. This is not surprising, as it is one of the few known perfectly secret encryption schemes.

The main difference between the two, is that the spaces we work over are the multiplicative group $(\tilde{O}_{\mathbf{A}}, \cdot)$ whose identity is $\mathbf{I}_N$ in Theorem **??**, and the additive group $((\mathbb{Z}_2)^\ell, +)$ in the OTP; whose identity is the zero vector of length ℓ.

As in the OTP, we make the assumption that $\mathcal{K}, \mathcal{M}, \mathcal{C}$ are all equal to the group we are working over; $\tilde{O}_{\mathbf{A}}$, which it is closed under multiplication. In the OTP, a message is revealed by applying the key on the ciphertext: if $c = m \oplus k$ for k drawn from $\mathcal{K}$, then $c \oplus k = m$. Analogously here, for $\mathbf{\Pi}$ drawn from $\tilde{O}_{\mathbf{A}}$: if $\bar{\mathbf{C}} = \mathbf{\Pi} \cdot \mathbf{U_A}$, then $\bar{\mathbf{C}}^T \cdot \mathbf{\Pi} = (\mathbf{U_A}^T \cdot \mathbf{\Pi}^T) \cdot \mathbf{\Pi} = \mathbf{U_A}^T$. An important difference here is that the multiplication is not commutative.

Also, for two distinct messages m_0, m_1 which are encrypted with the same key k to c_0, c_1 respectively, it follows that $c_0 \oplus c_1 = m_1 \oplus m_2$ which reveals the XOR of the two messages. In our case, for the bases $\mathbf{U}_0, \mathbf{U}_1$ encrypted to $\mathbf{C}_0 = \mathbf{\Pi}\mathbf{U}_0$ and $\mathbf{C}_0 = \mathbf{\Pi}\mathbf{U}_1$ with the same projection matrix $\mathbf{\Pi}$, it follows that $\mathbf{C}_0^T \cdot \mathbf{C}_1 = \mathbf{U}_0^T \cdot \mathbf{U}_1$.

APPENDIX E
EXTENSION TO OTHER OPERATIONS AND SCHEMES

In this appendix, we discuss how applying a random projection $\mathbf{\Pi}$ can be utilized in existing CC schemes, both approximate and exact, to securely recover other matrix operations. The main idea is that after we apply and arbitrary $\mathbf{\Pi}$ to the underlying matrix or matrices, the analysis and corresponding conclusion of Theorem **??** still applies. Once the information is encrypted through $\mathbf{\Pi}$, *e.g.* $\grave{\mathbf{A}} = \mathbf{\Pi}\mathbf{A}, \grave{\mathbf{b}} = \mathbf{\Pi}\mathbf{b}$, we then carry out the CC of choice, and we will recover the same result as if no encryption took place, without requiring an additional decryption step for the least squares problem and matrix multiplication, and does not increasing the system's redundancy. The drawback of this approach is the additional encryption step which corresponds to matrix multiplication. Fast matrix multiplication can be used to secure the data [**?**], [**?**], which is faster than computing $\mathbf{x}_{ls}^\star = \mathbf{A}^\dagger \mathbf{x}$.

We show how this approach is applied to GCSs for linear and logistic regression through SD, as well as *coded matrix multiplication* CMM schemes, and an approximate matrix inversion CC scheme; which is a non-linear operation [**?**]. In this scheme we utilize the structure of the gradient of the respective objective functions.

What was discuss in this section resembles *Homomorphic Encryption* [**?**], [**?**], [**?**], which allows computations to be performed over encrypted data; and has been used to enable privacy-preserving machine learning. Two main drawbacks of homomorphic encryption though is that it leads to many orders of magnitude slow down in training, and it allows collusion between a larger number of workers [**?**]. Moreover, the privacy guarantees rely on computational assumptions, while our approach is information-theoretic. Furthermore, we use orthogonal matrices for encrypting the data, which has been studied in the context of image-processing and message encryption [**?**], [**?**], [**?**], [**?**], [**?**], [**?**].

APPENDIX F
ORTHONORMAL ENCRYPTION FOR DISTRIBUTIVE TASKS

A. Securing Linear Regression

As pointed out in **??** and (**??**), for the modified objective

$$L_{\mathbf{\Pi}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]}) := \|\mathbf{\Pi}(\mathbf{Ax} - \mathbf{b})\|_2^2 = \|\grave{\mathbf{A}}\mathbf{x} - \grave{\mathbf{b}}\|_2^2 \,,$$

we have $\nabla_{\mathbf{x}} L_{\mathbf{\Pi}}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]}) = \nabla_{\mathbf{x}} L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[t]})$ for all t, *i.e.* $\hat{g}^{[t]} = g_{ls}^{[t]}$. It is clear that for $\mathbf{\Pi}$ an orthonormal matrix, there is no need to reverse the transformation to uncover the partial gradients.

Consider any GCS; exact or approximate, *e.g.* [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**]. If the workers are given partitions of $(\grave{\mathbf{A}}, \grave{\mathbf{b}})$, they will have no knowledge of $(\mathbf{A}, \mathbf{b})$, unless they learn $\mathbf{\Pi}$. According to Theorem **??**, this is not a concern. The workers therefore locally recover the partial gradients they were assigned to, and perform the encoding of the GCS which is being deployed. Once these are encoded and communicated to the central server. Once the sufficient encodings are received, *i.e.* when the threshold recovery is met, the central server can then recover the gradient at the given iteration.

B. Securing Logistic Regression

Another widely used algorithm whose solution is accelerated though gradient methods is logistic regression, which yields a linear classifier [**?**]. Applying a random orthonormal matrix can secure the information when GCSs are used to solve logistic regression, though at each iteration the central server will have to apply two encryptions. At each iteration t, the workers are collectively given $\grave{\mathbf{A}} = \mathbf{\Pi}_1 \cdot \mathbf{A}$, $\grave{\mathbf{a}}_i = \mathbf{\Pi}_2 \cdot \begin{bmatrix} 1 & \mathbf{a}_i^T \end{bmatrix}^T$ and $\grave{\mathbf{x}}^{[t]} = \mathbf{\Pi}_2 \cdot \mathbf{x}^{[t]}$, for $\mathbf{\Pi}_1 \in \tilde{O}_N(\mathbb{R})$ and $\mathbf{\Pi}_2 \in \tilde{O}_{d+1}(\mathbb{R})$. The gradient update to be recovered is

$$\hat{g}^{[t+1]} = \grave{\mathbf{A}}^T(\boldsymbol{\mu} - \mathbf{b}) \ \text{ for } \ \mu_i = \Big(1 + \exp\big(-\overbrace{\langle \grave{\mathbf{x}}^{[t]}, \grave{\mathbf{a}}_i \rangle}^{\langle \mathbf{x}^{[t]}, [1\ \mathbf{a}_i^T] \rangle}\big)\Big)^{-1} .$$

Thus $\hat{g}^{[t+1]} = \mathbf{\Pi}_1^T \cdot g^{[t+1]}$, so we apply $\mathbf{\Pi}_1$ to recover $g^{[t+1]}$. In this problem, the labels $\mathbf{b}_i \in \{0, 1\}$ are not hidden, as nothing can be inferred from these alone.

C. Securing Matrix Multiplication

Consider the matrices $\mathbf{A}_1 \in \mathbb{R}^{L \times N}$, $\mathbf{A}_2 \in \mathbb{R}^{N \times M}$ whose multiplication is to be computed by a CMM scheme. For $\mathbf{\Pi} \in \tilde{O}_N(\mathbb{R})$ orthonormal, by carrying out the CMM scheme on $\grave{\mathbf{A}}_1 = \mathbf{A}_1 \cdot \mathbf{\Pi}$ and $\grave{\mathbf{A}}_2 = \mathbf{\Pi} \cdot \mathbf{A}_2$, we recover

$$\grave{\mathbf{A}}_1 \cdot \grave{\mathbf{A}}_2 = \mathbf{A}_1 \cdot (\mathbf{\Pi}^T \mathbf{\Pi}) \cdot \grave{\mathbf{A}}_2 = \mathbf{A}_1 \cdot \mathbf{A}_2 \, , \tag{45}$$

and a security guarantee analogous to Theorem **??** holds. This encryption is useful when $N \ll L, M$, as otherwise the cost of encrypting the two matrices could be higher than that of performing the multiplication.

D. Securing Distributive Matrix Inversion

In [**?**] a CC scheme was used to recover an approximation of the inverse of a matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, by requesting from the workers to approximate as part of their computation a subset of the optimization problems

$$\check{\mathbf{b}}_i = \arg \min_{\mathbf{b} \in \mathbb{R}^N} \left\{ \|\mathbf{A}\mathbf{b} - \mathbf{e}_i\|_2^2 \right\} \tag{46}$$

for all $i \in \mathbb{N}_N$, where $\{\mathbf{e}_i\}_{i=1}^N$ is the standard basis. The solutions $\{\check{\mathbf{b}}\}_{i=1}^N$ comprise the columns of the inverse's estimate $\check{\mathbf{A}}^{-1}$, *i.e.*, $\mathbf{A}^{-1} \approx \check{\mathbf{A}}^{-1} = \begin{bmatrix} \check{\mathbf{b}}_1 & \cdots & \check{\mathbf{b}}_N \end{bmatrix}$, as

$$\mathbf{I}_N = \mathbf{A}\mathbf{A}^{-1} \approx \mathbf{A}\check{\mathbf{A}}^{-1} = \mathbf{A}\begin{bmatrix} \check{\mathbf{b}}_1 & \cdots & \check{\mathbf{b}}_N \end{bmatrix} = \begin{bmatrix} \mathbf{A}\check{\mathbf{b}}_1 & \cdots & \mathbf{A}\check{\mathbf{b}}_N \end{bmatrix} \, .$$

In this scheme, each worker has knowledge of the entire matrix $\mathbf{A}$. In our approach, instead of sharing $\mathbf{A}$ we share $\grave{\mathbf{A}} := \mathbf{A} \cdot \mathbf{\Pi}^T$, for a randomly chosen $\mathbf{\Pi} \in \tilde{O}_N(\mathbb{R})$. The workers then approximate

$$\check{\mathbf{b}}_i = \arg \min_{\mathbf{b} \in \mathbb{R}^N} \left\{ \|\grave{\mathbf{A}}\mathbf{b} - \mathbf{e}_i\|_2^2 \right\} \, , \tag{47}$$

thus $\grave{\mathbf{A}}^{-1} = \mathbf{\Pi} \cdot \check{\mathbf{A}}^{-1} = \begin{bmatrix} \check{\mathbf{b}}_1 & \cdots & \check{\mathbf{b}}_N \end{bmatrix}$.

As in the case of logistic regression, here we also need an additional decryption step: $\mathbf{\Pi}^T \cdot (\mathbf{\Pi} \cdot \check{\mathbf{A}}^{-1}) = \check{\mathbf{A}}^{-1}$ at the end of the process, to recover the approximation.