Orthonormal Sketches for Secure Coded Regression

Neophytos Charalambides[♮], **Hessam Mahdavifar**[♮], **Mert Pilanci**[♯], **and Alfred O. Hero III**[♮]
[♮]EECS Department University of Michigan [♯]EE Department Stanford University
Email: neochara@umich.edu, hessam@umich.edu, pilanci@stanford.edu, hero@umich.edu

***Abstract*—In this work, we propose a method 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 in *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. We focus on the special case of the *Subsampled Randomized Hadamard Transform*, which we generalize to block sampling; and discuss how it can be used to secure the data.**

I. Introduction and Preliminaries

We propose a method to securely speed up linear regression by simultaneously leveraging random projections and distributed computations. Random projections are a classical way of performing dimensionality reduction, and are widely used in algorithmic and learning contexts [2]–[5]. 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 [6]–[18], and fall under the framework of *coded computing* (CC). Data security is also an increasingly important issue in CC [19]. We present our results in terms of the standard CC model [6], though they extend to any centralized distributed network; comprised of a central server and computational workers.

We focus on iterative sketching for steepest descent (SD) in the context of solving overdetermined linear systems. We propose to apply a random orthonormal projection before distributing the data, and then perform mini-batch stochastic steepest descent (SSD) distributively on the transformed system. A special case of such a projection is the *Subsampled Randomized Hadamard Transform* (SRHT) [4], which relates to the *fast Johnson-Lindenstrauss transform* [20], [21]. 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.

We note that in CC, the workers are assumed to be homogeneous and all are assumed to have the same expected response time. In the proposed method, we stop receiving computations once a fixed fraction of workers respond, which results in a different 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 [22]–[36]. These schemes are collectively called *gradient coding* (GC). We note that iterative sketching has proven to be a powerful tool for second-order methods [37], [38], though it has not been explored in first-order methods. Since we consider a modified problem at each iteration, the method we propose is an *approximate* GC scheme. Related approaches have been proposed in [27]–[36]. Two benefits of our approach are that we do not require a decoding step, nor an encoding step by the workers; at each iteration.

Another benefit of our proposed 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.

There are related works to what we propose. The work of [39] focuses on parameter averaging for variance reduction, but only mentions a security guarantee for the Gaussian sketch, derived in [40]. Another line of work is that of [41], [42], which focuses on introducing redundancy through equiangular tight frames (ETFs), and partitioning the system into smaller linear systems, and then averaging the solutions of a fraction of them. A drawback of using ETFs, is the fact that most of them are over $\mathbb{C}$. The authors of [43] study privacy of random projections, though make the assumption that the projections meet the 'ε-MI-DP constraint'. Lastly, a secure GC scheme is studied in [44], though this work does not utilize sketching.

The paper is organized as follows. In II we review the framework and background for coded linear regression, and the ℓ_2-subspace embedding property. In III we present the proposed sketching algorithm, and in IV the special case where the projection is the Hadamard transform, which we refer to as *block-SRHT*. In V we present the security guarantee of our algorithm, and the modified version of the block-SRHT; which guarantees computational security. Finally, we present numerical experiments in VI; and concluding remarks in VII.

II. Coded Linear Regression

A. Least Squares Approximation and Steepest Descent

In linear least squares approximation [4], we approximate

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

This work was partially supported by grant ARO W911NF-15-1-0479.
All proofs can be found online in [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 i^{th} sample, i.e. $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_N\end{bmatrix}^T$ and $\mathbf{b} = \begin{bmatrix}b_1 & \cdots & b_N\end{bmatrix}^T$.

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

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

followed by updating the parameter vector: $\mathbf{x}^{[t+1]} \leftarrow \mathbf{x}^{[t]} - \xi_t \cdot g_{ls}^{[t]}$. The step-size $\xi_t > 0$ is determined by the central server. The exponent $[t]$ indexes the iteration $t = 1, 2, 3, ...$ which we drop when it is clear from the context.

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 (1), the data is partitioned as

$$\mathbf{A} = \begin{bmatrix}\mathbf{A}_1^T & \cdots & \mathbf{A}_K^T\end{bmatrix}^T \quad \text{and} \quad \mathbf{b} = \begin{bmatrix}\mathbf{b}_1^T & \cdots & \mathbf{b}_K^T\end{bmatrix}^T \tag{2}$$

where $\mathbf{A}_i \in \mathbb{R}^{\tau\times d}$ and $\mathbf{b}_i \in \mathbb{R}^\tau$ for all i, and $\tau = N/K$. To simplify the presentation, 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.

We denote the row vectors of a matrix $\mathbf{M}$ by $\mathbf{M}_{(i)}$, and the column vectors by $\mathbf{M}^{(j)}$. Our embedding results are presented in terms of an arbitrary partition $\mathbb{N}_N = \sqcup_{\iota=1}^K \mathcal{K}_\iota$, for $\mathbb{N}_N \coloneqq \{1, \cdots, N\}$ the index set of $\mathbf{M}$'s rows. The notation $\mathbf{M}_{(\mathcal{K}_\iota)}$ denotes the submatrix of $\mathbf{M}$ comprised of the rows indexed by $\mathcal{K}_\iota$. That is: $\mathbf{M}_{(\mathcal{K}_\iota)} = \mathbf{I}_{(\mathcal{K}_\iota)} \cdot \mathbf{M}$, for $\mathbf{I}_{(\mathcal{K}_\iota)}$ the corresponding submatrix of $\mathbf{I}_N$. We call $\mathbf{M}_{(\mathcal{K}_\iota)}$ the 'ι^{th} block of $\mathbf{M}$'.

In GC [22], the servers encode their computations $g_i \coloneqq \nabla_{\mathbf{x}} L_{ls}(\mathbf{A}_i, \mathbf{b}_i; \mathbf{x})$; which are then communicated to the central server. We refer to g_i's as *partial gradients*. Once a certain fraction of encoded partial gradients 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 is carried out 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 [25].

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

$$\hat{\mathbf{x}}_{ls} = \arg\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ L_{\mathbf{S}^{[t]}}(\mathbf{A}, \mathbf{b}; \mathbf{x}) \coloneqq \|\mathbf{S}^{[t]}(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2^2 \right\} \tag{3}$$

for $\mathbf{S}^{[t]} \in \mathbb{R}^{r\times N}$ a sketching matrix, with $r < N$. This is the core idea behind our approximation, where we incorporate iterative sketching with orthonormal matrices for $\mathbf{S}^{[t]}$, for our GC approach. The projection, is also what provides security against the workers and eavesdroppers.

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 (3) 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.

C. The ℓ_2-Subspace Embedding Property

For the analysis of the sketching matrices $\mathbf{S}_{\mathbf{\Pi}}$ we propose, we consider any orthonormal basis $\mathbf{U} \in \mathbb{R}^{N\times d}$ of the column-space of $\mathbf{A}$, i.e. $\text{im}(\mathbf{A}) = \text{im}(\mathbf{U})$.

Recall that the ℓ_2-*subspace embedding property* [3] states that any $\mathbf{y} \in \text{im}(\mathbf{U})$ satisfies:

$$\|\mathbf{I}_d - (\mathbf{S}_{\mathbf{\Pi}}\mathbf{U})^T(\mathbf{S}_{\mathbf{\Pi}}\mathbf{U})\|_2 \leqslant \epsilon \tag{4}$$

for $\epsilon > 0$. In turn, this characterizes the approximation's error of the solution $\hat{\mathbf{x}}_{ls}$ of (3) for $\mathbf{S} \leftarrow \mathbf{S}_{\mathbf{\Pi}}$, as

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

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

III. Block Subsampled Orthonormal Sketches

Sampling blocks (i.e. submatrices) 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 [45]–[48]. 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 considered as stragglers and disregarded. The proposed sketching matrices are summarised in Algorithm 1.

Algorithm 1: Subsampled Orthonormal Sketches

Input: $\mathbf{A} \in \mathbb{R}^{N\times d}$, $\tau = \frac{N}{K}$, and $q = \frac{r}{\tau} > \frac{d}{\tau}$
Output: sketching matrix $\mathbf{S}_{\mathbf{\Pi}} \in \mathbb{R}^{r\times N}$, sketch $\widetilde{\mathbf{A}}_r \in \mathbb{R}^{r\times d}$
Initialize: $\mathbf{\Omega} = \mathbf{0}_{q\times K}$
Randomly Select: $\mathbf{\Pi} \in O_N(\mathbb{R})$, an orthonormal matrix
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
$\mathbf{\Omega}_q \leftarrow \mathbf{\Omega} \otimes \mathbf{I}_\tau$
$\mathbf{S}_{\mathbf{\Pi}} \leftarrow \mathbf{\Omega}_q \cdot \mathbf{\Pi}$
$\widetilde{\mathbf{A}}_r \leftarrow \mathbf{S}_{\mathbf{\Pi}} \cdot \mathbf{A}$

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

A. Distributed Steepest Descent and Iterative Sketching

We now discuss the workers' computational tasks of our proposed GC scheme, 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 $\mathbf{G} \coloneqq \sqrt[]{N/r} \cdot \mathbf{\Pi}$, which are then partitioned into K blocks $(\tilde{\mathbf{A}}_i, \tilde{\mathbf{b}}_i)$; similar to (2), and distributed to the workers. Specifically, $\tilde{\mathbf{A}}_i = \mathbf{I}_{(\mathcal{K}_i)} \cdot \tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}_i = \mathbf{I}_{(\mathcal{K}_i)} \cdot \tilde{\mathbf{b}}$. This differs to most GC schemes, 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 multiset of the first q responsive workers is $\mathcal{S}^{[t]}$, the aggregated gradient

$$\hat{g}^{[t]} = 2 \cdot \sum_{j \in \mathcal{S}^{[t]}} \tilde{\mathbf{A}}_j^T \left(\tilde{\mathbf{A}}_j \mathbf{x}^{[t]} - \tilde{\mathbf{b}}_j \right) \tag{5}$$

is equal to the gradient of $L_{\mathbf{S}}$ for $\mathbf{S} \leftarrow \mathbf{S}_{\mathbf{\Pi}}^{[t]}$ the induced sketching matrix at each 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 $\mathbf{\Omega}_q^{[t]}$ and index set $\mathcal{S}^{[t]}$, correspond to the q responsive workers.

In Algorithm 1 and Theorems 5 and 6, we assume sampling uniformly with replacement. In what we just described, we used one replica of each block, thus $K = m$. To compensate for this, more than one replicas of each block could be distributed. This is not a major concern with uniform sampling, as the probability that the i^{th} block would be sampled more than once is $(q-1)/K^2$, which is negligible for large K. Furthermore, we sample *uniformly* at random in Algorithm 1, as the application of $\mathbf{\Pi}$ flattens the *block-leverage scores* [36], [49], i.e. they are all approximately equal. That is, for $\tilde{\mathbf{V}} = \mathbf{\Pi}\mathbf{U}$, we have $\tilde{\ell}_\iota \coloneqq \|\tilde{\mathbf{V}}_{(\mathcal{K}_\iota)}\|_F^2 \approx \frac{d}{K}$ for all $\iota \in \mathbb{N}_K$.

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

By Lemma 1, the Gram matrix of $\mathbf{S}_{[t]}$ in expectation satisfies the subspace embedding identity (4) with $\epsilon = 0$, as $\mathbb{E}\left[\mathbf{U}^T \cdot (\mathbf{S}_{[t]}^T \mathbf{S}_{[t]}) \cdot \mathbf{U}\right] = \mathbf{U}^T \mathbb{E}\left[\mathbf{S}_{[t]}^T \mathbf{S}_{[t]}\right] \mathbf{U} = \mathbf{U}^T \mathbf{U} = \mathbf{I}_d$.

Theorem 2. *The proposed GC scheme results in a mini-batch stochastic steepest descent procedure for*

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ L_{\mathbf{G}}(\mathbf{A}, \mathbf{b}; \mathbf{x}) \coloneqq L_{ls}(\mathbf{G}\mathbf{A}, \mathbf{G}\mathbf{b}; \mathbf{x}) \right\}. \tag{6}$$

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

Lemma 3. *The optimal solution of the modified least squares problem on $L_{\mathbf{G}}$, is equal to the optimal solution $\mathbf{x}_{ls}^\star$ of* (1)*.*

Note that $\mathbb{E}\left[\hat{g}^{[t]}\right] = \frac{q}{K} \cdot g_{ls}^{[t]}$ means the estimate $\hat{g}^{[t]}$ is unbiased after an appropriate rescaling. This rescaling could be incorporated in the step-size ξ_t. The subsampling which takes place; as a consequence of considering the q fastest responses, is the reason the distributive procedure results in a SSD approach for the modified problem (6). By Theorem 2 and Lemma 3, 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 $\mathcal{O}(1/\sqrt{t} + r/t)$ [50].

Corollary 4. *Consider the problems* (1) *and* (6)*, which are respectively solved through SD and our iterative sketching based GC scheme. Assume that the two approaches have the same starting point $\mathbf{x}^{[0]}$ and index set $\mathcal{S}^{[t]}$ at each t; and $\tilde{\xi}_t = \frac{K}{q} \cdot \xi_t$ the step-sizes used for our scheme. Then, in expectation, our scheme 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 3 and Corollary 4, our iterative sketching scheme approaches the optimal solution of the original problem (1), by solving the modified regression problem (6). Next, we present our main ℓ_2-subspace embedding result.

Theorem 5. *Fix $\epsilon > 0$ such that $\epsilon \ll 1/N$. Then, the sketching matrix $\mathbf{S}_{\mathbf{\Pi}}$ of Algorithm 1 is a $(1 \pm \epsilon)$-embedding of $\mathbf{A}$, according to* (4)*. Specifically, for $\delta > 0$ and $q = \Theta(\frac{d}{\tau} \log(2d/\delta)/\epsilon^2)$:*

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

IV. The Block-SRHT

In this section, we focus on a special case of $\mathbf{\Pi}$ which can be utilized in Algorithm 1, the randomized Hadamard transform. 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. In the new basis the block-leverage scores are close to uniform, hence uniform sampling is applied to reduce the effective dimension N, whilst the information of the data matrix 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 1. We give a subspace embedding guarantee for the block-wise sampling version or SRHT, which characterizes the approximation of our proposed GC for linear regression.

We refer to this special case as the "block-SRHT", for which $\mathbf{\Pi}$ is taken from the set of orthonormal matrices

$$H_N \coloneqq \left\{ \hat{\mathbf{H}}_N \cdot \mathbf{D} \ : \ \mathbf{D} = \mathrm{diag}(\pm 1) \in \{0, \pm 1\}^{N \times N} \right\}, \tag{7}$$

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

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

The SRHT introduced in [4] corresponds to the case where we select $\tau = 1$, i.e. $K = N$. The main differences in $\mathbf{S}_{\mathbf{\Pi}}$ is the sampling matrix $\mathbf{\Omega}_q$, and that $q = r/\tau$ sampling trials take place instead of r. Henceforth, we drop the subscript N. The limiting computational step in applying $\mathbf{S}_{\mathbf{\Pi}}$ in (3) is the multiplication by $\hat{\mathbf{H}}$. The recursive structure of $\hat{\mathbf{H}}$ permits us to compute $\mathbf{S}_{\mathbf{\Pi}} \cdot \mathbf{A}$ in $\mathcal{O}(Nd \log N)$ time, by using Fourier based methods. Furthermore, the transformation $\hat{\mathbf{H}}\mathbf{D}$ also permits for

a very sparse random projection to be applied, instead of $\mathbf{\Omega}_q$ [20]. Also note that the diagonal entries of $\mathbf{D}$ is the only place in which randomness takes place other than the sampling.

In Theorem 6, we state our ℓ_2-subspace embedding result regarding the block-SRHT.

Theorem 6. *The block-SRHT $\mathbf{S}_{\mathbf{\Pi}}$ is a $(1\pm\epsilon)$-embedding of $\mathbf{A}$. For $\delta > 0$ and $q = \Theta\big(\frac{d}{\tau}\log(Nd/\delta)\cdot\log(2d/\delta)/\epsilon^2\big)$:*

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

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

V. Security of Orthonormal Sketches

In this section, we discuss the security of the proposed orthonormal-based sketching matrices, and that of the block-SRHT. The main idea behind securing the resulting sketches is that there are infinitely many options of $\mathbf{\Pi}$ 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 1 and the data matrix $\mathbf{A}$. First, we recall the definition of a perfectly secret cryptographic scheme.

Definition 7 (Ch.2 [51])**.** *A security scheme* Enc *with message, ciphertext and key spaces $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{K}$ respectively is* ***Shannon/perfectly secret*** *w.r.t. a probability distribution D over $\mathcal{M}$, if for all $\bar{m}\in\mathcal{M}$ and all $\bar{c}\in\mathcal{C}$:*

$$\Pr_{\substack{m\leftarrow D\\ k\leftarrow\mathcal{K}}}[m=\bar{m}\mid \mathsf{Enc}_k(m)=\bar{c}] = \Pr_{m\leftarrow D}[m=\bar{m}]\ , \tag{8}$$

which is equivalent to the condition that for all $m_0, m_1\in\mathcal{M}$:

$$\Pr_{k\leftarrow\mathcal{K}}[\mathsf{Enc}_k(m_0)=\bar{c}] = \Pr_{k\leftarrow\mathcal{K}}[\mathsf{Enc}_k(m_1)=\bar{c}]\ . \tag{9}$$

For an information-theoretic security guarantee, $\mathcal{M}$ 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 $\mathbf{A}$. 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 $\mathbf{A}$ be drawn from $\mathcal{M}$ w.r.t. D. We consider D to be the uniform distribution. Furthermore, an inherent limitation of Shannon secrecy is that $|\mathcal{K}|\geqslant|\mathcal{M}|$.

Theorem 8. *In Algorithm 1, 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 8 does not apply to the block-SRHT, as in this case it is restrictive to assume that $\mathbf{U}_{\mathbf{A}}\in H_N$. A simple computation on a specific example also shows that this sketching approach does not provide Shannon secrecy. 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

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

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 9. *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 $\tilde{\mathbf{H}}\coloneqq\mathbf{P}\hat{\mathbf{H}}\in\{\pm1/\sqrt{N}\}^{N\times N}$, and the new sketching matrix is

$$\mathbf{S}_{\tilde{\mathbf{\Pi}}} = \mathbf{\Omega}_q\cdot(\mathbf{P}\cdot\hat{\mathbf{H}})\cdot\mathbf{D} = \mathbf{\Omega}_q\cdot\tilde{\mathbf{H}}\cdot\mathbf{D} = \mathbf{\Omega}_q\cdot\tilde{\mathbf{\Pi}} \tag{10}$$

for which our flattening result still holds true (Corollary 10). The reason we "garble" $\hat{\mathbf{H}}$ is 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 N^2 options for the projection of $\hat{\mathbf{H}}\mathbf{D}$, while for $\tilde{\mathbf{\Pi}}=\tilde{\mathbf{H}}\mathbf{D}$ there are $N^2\cdot N! = \mathcal{O}(N^{1.5+N}e^{-N})$ options for the projection $\tilde{\mathbf{\Pi}}$. Moreover, for

$$\tilde{H}_N\coloneqq\{\mathbf{P}\cdot\mathbf{\Pi}\ :\ \mathbf{P}\in S_N \text{ and } \mathbf{\Pi}\in H_N\} \tag{11}$$

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 $\mathbf{D}$.

By the following Corollary, the result of Theorem 6 also holds for the *garbled block-SRHT* (an analogous result is used to prove that the scores of $\hat{\mathbf{H}}\mathbf{D}\mathbf{A}$ are flattened). Thus, we can apply any $\tilde{\mathbf{\Pi}}$ from $\tilde{H}_N$ in Algorithm 1, and get a valid sketch.

Corollary 10. *For $\mathbf{y}\in\mathbb{R}^N$ a fixed (orthonormal) column vector of $\mathbf{U}$, and $\mathbf{D}\in\{0,\pm1\}^{N\times N}$ with random equiprobable diagonal entries of ±1, we have:*

$$\Pr\left[\|\tilde{\mathbf{H}}\mathbf{D}\cdot\mathbf{y}\|_\infty > C\sqrt{\log(Nd/\delta)/N}\right] \leqslant \frac{\delta}{2d} \tag{12}$$

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

Moreover, the flattening result also holds true for random projections $\mathbf{R}$ whose entries are rescaled Rademacher random variables, i.e. $\mathbf{R}_{ij}=\pm1/\sqrt{N}$ with equal probability. The advantage of this is that we have a larger set of projections

$$\tilde{R}_N\coloneqq\left\{\mathbf{R}\in\{\pm1/\sqrt{N}\}^{N\times N}:\Pr[\mathbf{R}_{ij}=+1/\sqrt{N}]=1/2\right\}$$

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 $|\tilde{H}_N|$. A drawback of applying such a projection is that it is much slower than its Hadamard-based counterpart.

Next, we provide a computationally secure guarantee for the garbled block-SRHT, i.e. for $\mathbf{S}_{\tilde{\mathbf{\Pi}}} \leftarrow \mathbf{\Omega}_q \cdot \tilde{\mathbf{\Pi}}$. The guarantee of Theorem 11 against computationally bounded adversaries, relies heavily on the assumption that one-way functions (OWFs) exist. Even though OWFs are minimal cryptographic objects, it is not known whether such functions exist [51]. 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.

Theorem 11. *Under the assumption that one-way permutations exist, the garbled sketching matrix $\mathbf{S}_{\tilde{\mathbf{\Pi}}} \leftarrow \mathbf{\Omega}_q \cdot \tilde{\mathbf{\Pi}}$ is computationally secure against polynomial-bounded adversaries.*

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 8 and 11, we know that under certain assumptions we can secure $\mathbf{A}$.

Since the projections are orthonormal, it follows 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, logistic regression. This resembles a homomorphic encryption scheme, but is by no means fully-homomorphic.

VI. Experiments

We compared our proposed distributed GC schemes to analogous approaches where the projection $\mathbf{\Pi}$ is a Gaussian sketch or a Rademacher random matrix. Our approach was found to outperform both of these sketching methods in terms of convergence and approximation error.

We also compared our approach with uncoded (regular) SD and SSD. 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 $\text{im}(\mathbf{A})$, to define $\mathbf{b}$. We considered $K = 100$ blocks, thus $\tau = 20$. Each experiment was carried out six times, and we report the average in our plot. For the experiments in Figure 1 we ran a total of 400 iterations, and varied ξ for each experiment by logarithmic factors of the optimal step-size $\xi^{\times} = 2/\sigma_{\max}(\mathbf{A})^2$. The effective dimension N was reduced to $r = 1000$ in all experiments.

In Figure 1 we show how the residual error $\|\mathbf{x}_{ls}^{\star} - \hat{\mathbf{x}}\|_2$ behaves, with different step-sizes. In the depicted simulation, we considered a sparse matrix $\mathbf{A}$. In analogous experiments where we considered a dense matrix, or a matrix drawn from a t-distribution, the behaviors were similar. In all cases, the order of the magnitude of the residual error was the same.

In Figure 2 we present the residual error at each iteration, in the case where $\mathbf{A}$ was drawn from a t-distribution. We considered a fixed step-size at $\xi = 10^2 \cdot \xi^{\times}$. It is evident, that our proposed sketches result in faster convergence of $\hat{\mathbf{x}}$ per iteration, than SD and SSD. Our approach also outperformed the scenario when a Gaussian projection was applied.

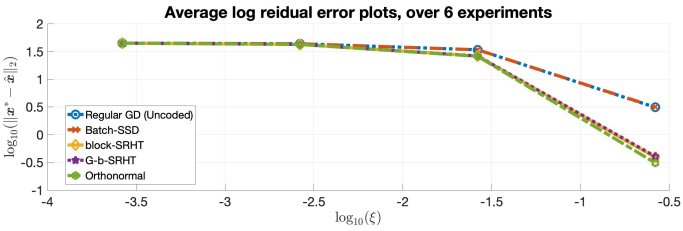

Fig. 1. log residual error, for $\mathbf{A}$ sparse.

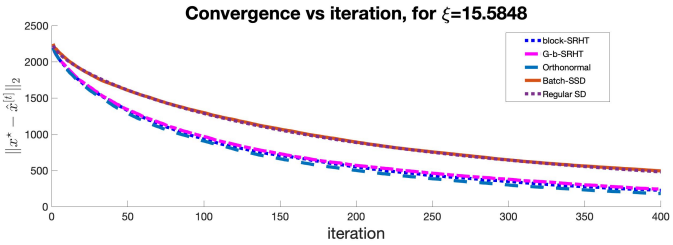

Fig. 2. Error at each iteration.

Lastly, we show the resulting block-leverage scores after applying the projections, in Figure 3. The flattening of these scores is precisely what permitted us to sample uniformly through $\mathbf{\Omega}_q$, and prove Theorems 5 and 6.

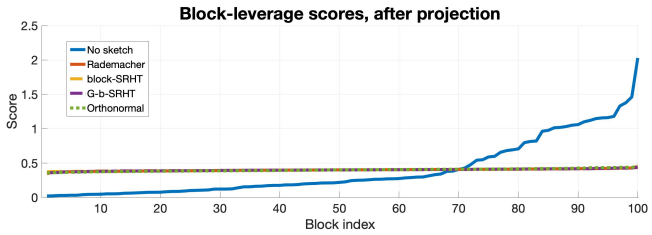

Fig. 3. Flattening of block-scores, for $\mathbf{A}$ following a t-distribution.

VII. Concluding Remarks

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 between this and other works is that the resulting sketches are sampling *blocks* uniformly, after applying random orthonormal projections. The 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 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 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 [14], the approximate result will be the same.

Moreover, our dimensionality reduction algorithm can be utilized by a single server, to store a very large data-matrix.

REFERENCES

[1] N. Charalambides, H. Mahdavifar, M. Pilanci, and A. O. Hero III, "Orthonormal Sketches for Secure Coded Regression," *arXiv preprint arXiv:2201.08522*, 2022.

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

[3] D. P. Woodruff, "Sketching as a tool for numerical linear algebra," *arXiv preprint arXiv:1411.4357*, 2014.

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

[5] P. Drineas and M. W. Mahoney, "RandNLA: Randomized Numerical Linear Algebra," *Communications of the ACM*, vol. 59, no. 6, pp. 80–90, 2016.

[6] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2017.

[7] A. Reisizadeh, S. Prakash, R. Pedarsani, and S. Avestimehr, "Coded computation over heterogeneous clusters," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2408–2412.

[8] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded distributed computing: Straggling servers and multistage dataflows," in *54th Annual Allerton Conference*. IEEE, 2016, pp. 164–171.

[9] K. Lee, C. Suh, and K. Ramchandran, "High-dimensional coded matrix multiplication," in *IEEE Int. Symp. Inf. Theory (ISIT)*. IEEE, 2017, pp. 2418–2422.

[10] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," in *Adv. in Neural Info. Proc. Systems (NIPS)*, 2016, pp. 2100–2108.

[11] A. Ramamoorthy, L. Tang, and P. O. Vontobel, "Universally decodable matrices for distributed matrix-vector multiplication," *arXiv preprint arXiv:1901.10674*, 2019.

[12] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security and privacy," *arXiv preprint arXiv:1806.00939*, 2018.

[13] M. Rudow, K. Rashmi, and V. Guruswami, "A locality-based approach for coded computation," *arXiv preprint arXiv:2002.02440*, 2020.

[14] N. Charalambides, M. Pilanci, and A. Hero, "Approximate Weighted CR-Coded Matrix Multiplication," *arXiv preprint arXiv:2011.09709*, 2020.

[15] N. Charalambides, M. Pilanci, and A. O. Hero III, "Straggler Robust Distributed Matrix Inverse Approximation," *arXiv preprint arXiv:2003.02948*, 2020.

[16] E. Ozfatura, S. Ulukus, and D. Gunduz, "Coded distributed computing with partial recovery," *arXiv preprint arXiv:2007.02191*, 2020.

[17] E. Ozfatura, B. Buyukates, D. Gunduz, and S. Ulukus, "Age-based coded computation for bias reduction in distributed learning," *arXiv preprint arXiv:2006.01816*, 2020.

[18] N. Charalambides, H. Mahdavifar, and A. O. Hero III, "Numerically stable binary coded computations," *arXiv preprint arXiv:2109.10484*, 2021.

[19] S. Li and S. Avestimehr, "Coded computing," *Foundations and Trends® in Communications and Information Theory*, vol. 17, no. 1, 2020.

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

[21] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," 1984.

[22] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *International Conference on Machine Learning*, 2017, pp. 3368–3376.

[23] W. Halbawi, N. Azizan, F. Salehi, and B. Hassibi, "Improving distributed gradient descent using Reed-Solomon codes," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 2027–2031.

[24] E. Ozfatura, D. Gunduz, and S. Ulukus, "Gradient coding with clustering and multi-message communication," *arXiv preprint arXiv:1903.01974*, 2019.

[25] N. Charalambides, H. Mahdavifar, and A. O. Hero, "Numerically stable binary gradient coding," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2622–2627.

[26] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," *arXiv preprint arXiv:1802.03475*, 2018.

[27] N. Raviv, I. Tamo, R. Tandon, and A. G. Dimakis, "Gradient coding from cyclic MDS codes and expander graphs," *arXiv preprint arXiv:1707.03858*, 2017.

[28] Z. Charles and D. Papailiopoulos, "Gradient coding via the stochastic block model," *arXiv preprint arXiv:1805.10378*, 2018.

[29] Z. Charles, D. Papailiopoulos, and J. Ellenberg, "Approximate gradient coding via sparse random graphs," *arXiv preprint arXiv:1711.06771*, 2017.

[30] H. Wang, Z. Charles, and D. Papailiopoulos, "Erasurehead: Distributed gradient descent without delays using approximate gradient coding," *arXiv preprint arXiv:1901.09671*, 2019.

[31] R. Bitar, M. Wootters, and S. El Rouayheb, "Stochastic gradient coding for flexible straggler mitigation in distributed learning."

[32] S. Wang, J. Liu, and N. Shroff, "Fundamental limits of approximate gradient coding," *arXiv preprint arXiv:1901.08166*, 2019.

[33] S. Kadhe, O. Ozan Koyluoglu, and K. Ramchandran, "Gradient coding based on block designs for mitigating adversarial stragglers," *arXiv preprint arXiv:1904.13373*, 2019.

[34] S. Horii, T. Yoshida, M. Kobayashi, and T. Matsushima, "Distributed stochastic gradient descent using ldgm codes," *arXiv preprint arXiv:1901.04668*, 2019.

[35] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "Draco: Byzantine-resilient distributed training via redundant gradients," *arXiv preprint arXiv:1803.09877*, 2018.

[36] N. Charalambides, M. Pilanci, and A. O. Hero, "Weighted gradient coding with leverage score sampling," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5215–5219.

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

[38] J. Lacotte, S. Liu, E. Dobriban, and M. Pilanci, "Optimal iterative sketching methods with the subsampled randomized hadamard transform," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[39] B. Bartan and M. Pilanci, "Distributed sketching methods for privacy preserving regression," *arXiv preprint arXiv:2002.06538*, 2020.

[40] S. Zhou, L. Wasserman, and J. Lafferty, "Compressed regression," in *Advances in Neural Information Processing Systems*, vol. 20, 2008.

[41] C. Karakus, Y. Sun, and S. Diggavi, "Encoded distributed optimization," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2890–2894.

[42] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Redundancy techniques for straggler mitigation in distributed optimization and learning," *Journal of Machine Learning Research*, vol. 20, no. 72, pp. 1–47, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-148.html

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

[44] Q. Yu and A. S. Avestimehr, "Harmonic coding: An optimal linear code for privacy-preserving gradient-type computation," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 1102–1106.

[45] T. Elfving, "Block-iterative methods for consistent and inconsistent linear equations," *Numerische Mathematik*, vol. 35, no. 1, pp. 1–12, 1980.

[46] M. H. Gutknecht, "Block krylov space methods for linear systems with multiple right-hand sides: an introduction," 2006.

[47] Needell, Deanna and Tropp, Joel A, "Paved with good intentions: analysis of a randomized block kaczmarz method," *Linear Algebra and its Applications*, vol. 441, pp. 199–221, 2014.

[48] E. Rebrova and D. Needell, "On block gaussian sketching for the kaczmarz method," *Numerical Algorithms*, pp. 1–31, 2020.
[49] U. Oswal, S. Jain, K. S. Xu, and B. Eriksson, "Block cur: Decomposing matrices using groups of columns," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 360–376.
[50] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches." *Journal of Machine Learning Research*, vol. 13, no. 1, 2012.
[51] J. Katz and Y. Lindell, *Introduction to modern cryptography*. Chapman and Hall/CRC, 2014.
[52] M. W. Mahoney, "Lecture notes on randomized linear algebra," *arXiv preprint arXiv:1608.04481*, 2016.
[53] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, "Fast approximation of matrix coherence and statistical leverage," *Journal of Machine Learning Research*, vol. 13, no. Dec, pp. 3475–3506, 2012.
[54] S. Wang, "A practical guide to randomized matrix computations with matlab implementations," *arXiv preprint arXiv:1505.07570*, 2015.

APPENDIX A
PROOFS OF SECTION III-A

Proof. [Lemma 1] The only difference in $\mathbf{S}_{\mathbf{\Pi}}^{[t]}$ at each iteration, is $\mathcal{S}^{[t]}$ and $\mathbf{\Omega}_q^{[t]}$. This corresponds to a uniformly 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 $\mathbf{\Omega}_q^{[t]}$. Let $\mathcal{Q}$ be the set of all subsets of $\mathbb{N}_K$ of size q. Then

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

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

Proof. [Theorem 2] The only difference in $\mathbf{S}_{\mathbf{\Pi}}^{[t]}$ at each iteration, is $\mathcal{S}^{[t]}$ and $\mathbf{\Omega}_q^{[t]}$. This corresponds to a uniformly 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 $\mathbf{\Omega}_q^{[t]}$. By (5), 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 $\mathbf{\Pi}\mathbf{U}$, 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

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

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 (1) 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. □

Proof. [Lemma 3] 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 (1), as

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

□

Proof. [Corollary 4] 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

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

which completes the inductive step. □

Next, we provide the proof of Theorem 6. First, we to present the key results regarding the leverage and block-leverage scores of $\mathbf{\Pi A}$ (Lemmas 12, 14).Throughout this subsection, by ℓ_i we denote the i^{th} leverage score of $\mathbf{\Pi A}$ for $\mathbf{\Pi}$ a random orthonormal matrix, i.e.

$$\ell_i = \|\tilde{\mathbf{U}}_{(i)}\|_2^2 = \|\mathbf{e}_i^T\tilde{\mathbf{U}}\|_2^2 = \mathbf{e}_i^T\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\mathbf{e}_i \tag{13}$$

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 i^{th} standard basis vector of $\mathbb{R}^N$.

Lemma 12. *For each $i \in \mathbb{N}_N$, we have $\mathbb{E}[\ell_i] = \frac{d}{N}$.*

Proof. By (13), we have

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

□

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

$$\grave{\ell}_\iota = \frac{1}{d}\cdot\|\mathbf{I}_{(\mathcal{K}_\iota)}\tilde{\mathbf{U}}\|_F^2 = \frac{1}{d}\cdot\Big(\sum_{j\in\mathcal{K}_\iota}\ell_j\Big) = \sum_{j\in\mathcal{K}_\iota}\bar{\ell}_j\ . \tag{14}$$

To prove Lemma our results, we first recall Hoeffding's inequality.

Theorem 13 (Hoeffding's Inequality, [52])**.** *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*

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

Lemma 14. *The normalized leverage scores $\{\bar{\ell}_i\}_{i=1}^N$ of $\mathbf{\Pi A}$ satisfy*

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

for any $\epsilon > 0$.

Proof. [Lemma 14] 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 12, 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 an $\epsilon > 0$. By applying Theorem 13, we get

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

thus

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

□

Lemma 15. *For all $\iota \in \mathbb{N}_K$ and $\mathcal{K}_\iota \subsetneq \mathbb{N}_N$ of size $\tau = N/K$*

$$\Pr\left[\big|\grave{\ell}_\iota - 1/K\big| < \tau\epsilon\right] = \Pr\left[\grave{\ell}_\iota <_{N\epsilon} 1/K\right] > 1 - 2\tau\cdot e^{-2\epsilon^2/N}$$

for $\epsilon > 0$.

Proof. [Lemma 15] By Lemma (14), it follows that

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

where in $\bowtie$ we applied the binomial approximation. □

The proof of Corollary 5 is a direct consequence of Lemma 15 and Theorem 16. We note that in our statement we make the assumption that $\grave{\ell}_\iota = 1/K$ for all ι, even though this is not necessarily the case, as Lemma 15 allows a small deviation. One could generalize Theorem 16 to accommodate sampling according to *approximate* block-leverage scores, e.g. [53]. This is not studied in our work.

Theorem 16. *The sketching matrix $\mathbf{S_\Pi}$ 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 $\sqrt{1/(q\grave{\ell}_\iota)}$, is a $(1\pm\epsilon)$-embedding of $\mathbf{A}$; as defined in* (4)*. Specifically, for $\delta > 0$ and $q = \Theta(\frac{d}{\tau}\log(2d/\delta)/\epsilon^2)$:*

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

APPENDIX B
PROOFS OF SECTION IV

In this appendix, we present two lemmas which we use to bound the entries of $\hat{\mathbf{V}} \coloneqq \hat{\mathbf{H}}\mathbf{DU}$, and its *leverage scores* $\ell_i \coloneqq \|\hat{\mathbf{V}}_{(i)}\|_2^2$, for which $\sum_{i=1}^N \ell_i = d$. Leverage scores induce a sampling distribution which has proven to be useful in linear regression [3], [52]–[54] and GC [36]. From these lemmas, we deduce that the leverage scores of $\hat{\mathbf{H}}\mathbf{DA}$ are close to being

uniform, implying that the *block-leverage scores* [36] are also uniform, which is precisely what Lemma 20 states.

Lemma 19 is a variant of the Flattening Lemma [20], [52], 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 [3, Fact 1], which we apply to prove our subspace embedding guarantees. Finally, we present proofs of Proposition 23 and Theorems 2, 6.

Lemma 17 (Azuma-Hoeffding Inequality, [52])**.** *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*

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

Theorem 18 (Matrix Chernoff Bound, [3, 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, $\forall \epsilon > 0$:*

$$\Pr\Big[\|\mathbf{Z}\|_2 > \epsilon\Big] \leqslant 2d\cdot\exp\left(-\frac{q\epsilon^2}{\sigma^2+\gamma\epsilon/3}\right). \tag{15}$$

Lemma 19 (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:*

$$\Pr\Big[\|\hat{\mathbf{H}}\mathbf{D}\cdot\mathbf{y}\|_\infty > C\sqrt{\log(Nd/\delta)/N}\Big] \leqslant \frac{\delta}{2d} \tag{16}$$

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

Proof. [Lemma 19] 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 17

$$\begin{aligned}\Pr\left[\Big|\sum_{j=1}^{N} Z_j\Big| > \rho\right] &\leqslant 2\exp\left\{\frac{-\rho^2/2}{\sum_{j=1}^{N}(\mathbf{y}_j/\sqrt{N})^2}\right\} \\ &= 2\exp\left\{\frac{-N\rho^2}{2\cdot\langle\mathbf{y},\mathbf{y}\rangle}\right\} \overset{\flat}{=} 2\cdot e^{-N\rho^2/2}\end{aligned} \tag{17}$$

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

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

where $\natural$ follows from the upper bound on C. By applying the union bound over all $i \in \mathbb{N}_N$, we attain (16). □

Lemma 20. *For all $i \in \mathbb{N}_N$ and $\{\mathbf{e}_i\}_{i=1}^N$ the standard basis:*

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

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

Proof. [Lemma 20] It is straightforward that the columns of $\hat{\mathbf{V}}$ form an orthonormal basis of $\mathbf{A}$, thus Lemma 19 implies that for $j \in \mathbb{N}_d$

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

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

$$\Pr\left[\overbrace{|\mathbf{e}_i^T\cdot\hat{\mathbf{V}}\cdot\mathbf{e}_j|}^{|(\hat{\mathbf{H}}\mathbf{D}\mathbf{U})_{ij}|} > C\sqrt{\frac{\log(Nd/\delta)}{N}}\right] \leqslant d\cdot\frac{\delta}{2d} = \delta/2\ . \tag{18}$$

We manipulate the argument of the above bound to obtain

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

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

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

and the proof is complete. □

Remark 21. *The complementary probable event of* (18) *can be interpreted as 'every entry of $\hat{\mathbf{V}}$ is small in absolute value'.*

Lemma 22. *For all $\iota \in \mathbb{N}_K$ and $\mathcal{K}_\iota \subsetneq \mathbb{N}_N$ of size $\tau = N/K$*

$$\Pr\Big[\tilde{\ell}_\iota \leqslant C^2 d\cdot\log(Nd/\delta)/K\Big] > 1-\tau\delta/2\ .$$

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

Proof. [Lemma 22] For $\alpha := C^2 d\cdot\log(Nd/\delta)/N$

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

where $\diamondsuit$ follows from Lemma 20. By the binomial approximation, we have $(1-\delta/2)^\tau \approx 1-\tau\delta/2$. □

Define the symmetric matrices

$$\mathbf{X}_i = \left(\mathbf{I}_d - \frac{N}{\tau}\cdot\hat{\mathbf{V}}_{(\mathcal{K}^i)}^T\hat{\mathbf{V}}_{(\mathcal{K}^i)}\right) = \left(\mathbf{I}_d - K\cdot\hat{\mathbf{V}}_{(\mathcal{K}^i)}^T\hat{\mathbf{V}}_{(\mathcal{K}^i)}\right) \tag{19}$$

where $\hat{\mathbf{V}}_{(\mathcal{K}^i)} = \hat{\mathbf{V}}_{(\mathcal{K}_\iota)}$ is the submatrix of $\hat{\mathbf{V}}$ corresponding to the i^{th} 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 (4). To apply Theorem 18, 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

$$\begin{aligned}
\|\mathbf{X}_i\|_2 &\leqslant \|\mathbf{I}_d\|_2 + \|\frac{N}{\tau}\cdot\hat{\mathbf{V}}_{(\mathcal{K}^i)}^T\hat{\mathbf{V}}_{(\mathcal{K}^i)}\|_2 \\
&= 1+\frac{N}{\tau}\cdot\|\hat{\mathbf{V}}_{(\mathcal{K}_\iota)}\|_2^2 \\
&\leqslant 1+\frac{N}{\tau}\cdot\max_{\iota\in\mathbb{N}_K}\left\{\|\mathbf{I}_{(\mathcal{K}_\iota)}\cdot\hat{\mathbf{V}}\|_2^2\right\} \\
&\leqslant 1+\frac{N}{\tau}\cdot\max_{\iota\in\mathbb{N}_K}\left\{\|\mathbf{I}_{(\mathcal{K}_\iota)}\cdot\hat{\mathbf{V}}\|_F^2\right\} \qquad [\|\mathbf{A}\|_2\leqslant\|\mathbf{A}\|_F] \\
&\overset{\$}{\leqslant} 1+\frac{N}{\tau}\cdot\left(|\mathcal{K}_\iota|\cdot\max_{j\in\mathbb{N}_N}\left\{\|\mathbf{e}_j^T\cdot\hat{\mathbf{V}}\|_2^2\right\}\right) \\
&\leqslant 1+\frac{N}{\tau}\cdot\left(\tau\cdot(C^2\cdot d\log(Nd/\delta)/N)\right) \text{ [Lemma 19]} \\
&= 1+C^2\cdot d\log(Nd/\delta) \qquad (20) \\
&= 1+N\alpha
\end{aligned}$$

for $\alpha = C^2 d\cdot\log(Nd/\delta)/N$ 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

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

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 $\hat{\mathbf{V}}$:

$$\|\hat{\mathbf{V}}_l\|_2^2 = \|\hat{\mathbf{V}}_l\hat{\mathbf{V}}_l^T\|_2 = \|\hat{\mathbf{V}}_l^T\hat{\mathbf{V}}_l\|_2 \leqslant \alpha \qquad (21)$$

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

Next, we compute $\mathbf{E} \coloneqq \mathbb{E}[\mathbf{X}^T\mathbf{X}+\mathbf{I}_d]$ 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:

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

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 $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. Note also that for all $\mathbf{y}\in\mathbb{R}^d$:

$$\begin{aligned}
\mathbf{y}^T\mathbf{E}\mathbf{y} &= K\cdot\mathbf{y}^T\left(\sum_{l=1}^N\hat{\mathbf{V}}_l^T\left(\hat{\mathbf{V}}_l\hat{\mathbf{V}}_l^T\right)\hat{\mathbf{V}}_l\right)\mathbf{y} \\
&\overset{\sharp}{=} K\cdot\sum_{l=1}^N\langle\mathbf{y},\hat{\mathbf{V}}_l\rangle^2\cdot\|\hat{\mathbf{V}}_l\|_2^2 \\
&\overset{\flat}{\leqslant} K\alpha\cdot\sum_{l=1}^N\langle\mathbf{y},\hat{\mathbf{V}}_l\rangle^2 \qquad (23) \\
&= K\alpha\cdot\sum_{l=1}^N\mathbf{y}^T\hat{\mathbf{V}}_l^T\cdot\hat{\mathbf{V}}_l\mathbf{y} \\
&= \mathbf{y}^T\left(K\alpha\cdot\sum_{l=1}^N\hat{\mathbf{V}}_l^T\cdot\hat{\mathbf{V}}_l\right)\mathbf{y} \\
&= \mathbf{y}^T\mathbf{F}\mathbf{y}
\end{aligned}$$

where in $\flat$ we invoked (21). 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

$$\mathbf{w}_i^T\left(\mathbf{Q}_\mathbf{E}\mathbf{\Lambda}_\mathbf{E}\mathbf{Q}_\mathbf{E}^T\right)\mathbf{w}_i = \mathbf{e}_i^T\cdot\mathbf{\Lambda}_\mathbf{E}\cdot\mathbf{e}_i = \lambda_i$$

and by (23) 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

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

In turn, this gives us

$$\begin{aligned}
\|\mathbb{E}[\mathbf{X}^T\mathbf{X}]\|_2 &= \|\mathbf{E}-\mathbf{I}_d\|_2 \\
&\leqslant \|\mathbf{E}\|_2+\|\mathbf{I}_d\|_2 \\
&\leqslant \|\mathbf{F}\|_2+1 \\
&= K\alpha+1 \\
&\leqslant C^2K\frac{d}{N}\log(Nd/\delta)+1 \\
&= C^2\frac{d}{\tau}\log(Nd/\delta)+1 \qquad (24)
\end{aligned}$$

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 18.

Proposition 23. *The block-SRHT $\mathbf{S_\Pi}$ guarantees*

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

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

Proof. [Proposition 23] Let $\{\mathbf{X}_i\}_{i=1}^q$ as defined in (19) denote q block samples. Let $j(i)$ denote the index of the submatrix which was sampled at the i^{th} random trial, i.e. $\mathcal{K}_{j(i)} = \mathcal{K}^i_{j(i)}$. We then get

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

We apply Lemma 18 by fixing the terms we bounded: (20) $\gamma = C^2 d\log(Nd/\delta)+1$, (24) $\sigma^2 = C^2\frac{d}{\tau}\log(Nd/\delta)+1$, and fix q and ϵ. The denominator of the exponent in (15) is then

$$\begin{aligned}
&\big(C^2 d/\tau\cdot\log(Nd/\delta)+1\big) + \big((C^2 d\log(Nd/\delta)+1)\cdot\epsilon/3\big) = \\
&\quad = C^2 d/\tau\cdot\log(Nd/\delta)\cdot\big(1+\epsilon\tau/3\big) + (1+\epsilon/3) \\
&\quad = \Theta\left(\frac{d}{\tau}\log(Nd/\delta)\right)
\end{aligned}$$

and the proof is complete. □

Proof. [Theorem 6] By substituting q in the bound of Proposition 23 and taking the complementary event, we attain the statement. □

Appendix C
Proofs of Section V

Proof. [Theorem 8] Denote the application of $\mathbf{\Pi}$ to a matrix $\mathbf{M}$ by $\mathsf{Enc}_{\mathbf{\Pi}}(\mathbf{M}) = \mathbf{\Pi}\mathbf{M}$. We will prove secrecy of this scheme, which then implies that a subsampled version of the transformed information is also secure. Let $\grave{\mathbf{A}} = \mathsf{Enc}_{\mathbf{\Pi}}(\mathbf{A})$ and $\grave{\mathbf{b}} = \mathsf{Enc}_{\mathbf{\Pi}}(\mathbf{b})$.

The adversaries' goal is to reveal $\mathbf{A}$. To prove that $\mathsf{Enc}_{\mathbf{\Pi}}$ 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 $\mathbf{\Pi}^{-1}$, as the only way to recover $\mathbf{A}$ from $\grave{\mathbf{A}}$ is by inverting the transformation of $\mathbf{\Pi}$: $\mathbf{A} = \mathbf{\Pi}^{-1}\cdot\grave{\mathbf{A}}$. This contradicts the fact that only $(\grave{\mathbf{A}}, \grave{\mathbf{b}})$ were observed. Thus, $\mathsf{Enc}_{\mathbf{\Pi}}$ is a well-defined security scheme.

It remains to prove perfect secrecy according to Definition 7. Observe that for any $\bar{\mathbf{U}} \in \mathcal{M}$ and $\bar{\mathbf{Q}} \in \mathcal{C}$

$$\Pr_{\mathbf{\Pi}\leftarrow\mathcal{K}}\left[\mathsf{Enc}_{\mathbf{\Pi}}(\bar{\mathbf{U}}) = \bar{\mathbf{Q}}\right] = \Pr_{\mathbf{\Pi}\leftarrow\mathcal{K}}\left[\mathbf{\Pi}\cdot\bar{\mathbf{U}} = \bar{\mathbf{Q}}\right] = \tag{25}$$

$$= \Pr_{\mathbf{\Pi}\leftarrow\mathcal{K}}\left[\mathbf{\Pi} = \bar{\mathbf{Q}}\cdot\bar{\mathbf{U}}^{-1}\right] \overset{\sharp}{=} \frac{1}{|\tilde{O}_{\mathbf{A}}|} = \frac{1}{|\mathcal{K}|} \tag{26}$$

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

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

as required by Definition 7. This completes the proof. □

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

$$\grave{\mathbf{A}} = (\mathbf{\Pi}\cdot\mathbf{U}_{\mathbf{A}})\cdot\mathbf{\Sigma}_{\mathbf{A}}\cdot\mathbf{V}_{\mathbf{A}}^T = \mathbf{U}_{\grave{\mathbf{A}}}\cdot\mathbf{\Sigma}_{\mathbf{A}}\cdot\mathbf{V}_{\mathbf{A}}^T\ . \tag{27}$$

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 $\mathbf{U}_{\grave{\mathbf{A}}} = \mathbf{\Pi}\cdot\mathbf{U}_{\mathbf{A}}$.

Geometrically, the encoding $\mathsf{Enc}_{\mathbf{\Pi}}$ changes the orthonormal basis of $\mathbf{U}_{\mathbf{A}}$ to $\mathbf{U}_{\grave{\mathbf{A}}}$, by rotating it or reflecting it; when $\det(\mathbf{\Pi})$ 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. $\Sigma_{\mathbf{A}}$ 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 $\mathbf{\Pi}$.

Proof. [Corollary 10] The proof is identical to that of Lemma 19. 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 (17) holds true, the guarantees implied by flattening lemma also do, thus the sketching properties of the SRHT are maintained. □

Remark 24. *Since the Lemma 19 and Corollary 10 give the same result for the block-SRHT and garbled block-SRHT respectively, it follows that Theorem 6 also holds for the garbled block-SRHT.*

Definition 25 (Ch.3 [51])**.** *A security 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.*

Proof. [Theorem 11] Assume w.l.o.g. that a computationally bounded adversary observes $\widetilde{\mathbf{\Pi}}\mathbf{A}$, for which $\widetilde{\mathbf{A}}_r = \mathbf{S_\Pi}\cdot\mathbf{A} =$

$\mathbf{\Omega}_q \cdot (\tilde{\mathbf{\Pi}}\mathbf{A})$ is the resulting sketch of Algorithm 1, for $\tilde{\mathbf{\Pi}} \in \tilde{H}_N$. To invert the transformation of $\tilde{\mathbf{\Pi}}$, the adversary needs knowledge of the components of $\tilde{\mathbf{\Pi}}$, 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 $\tilde{\mathbf{\Pi}}\mathbf{A}$. This means that it has revealed $\mathbf{P}$, so that it can compute

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

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

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

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

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 7. 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 $\mathbf{\Pi}$.

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

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

which contradicts (9).

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 8, bares a strong resemblance with the one-time-pad (OTP). 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 8, and the additive group $\left((\mathbb{Z}/2\mathbb{Z})^\ell, +\right)$ 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 $\bar{\mathbf{C}} = \mathbf{\Pi} \cdot \mathbf{U}_{\mathbf{A}}$, then $\bar{\mathbf{C}}^T \cdot \mathbf{\Pi} = (\mathbf{U}_{\mathbf{A}}^T \cdot \mathbf{\Pi}^T) \cdot \mathbf{\Pi} = \mathbf{U}_{\mathbf{A}}^T$. 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 $\mathbf{C}_0^T \cdot \mathbf{C}_1 = \mathbf{U}_0^T \cdot \mathbf{U}_1$.