Distributed Local Sketching for ℓ_2 Embeddings

Neophytos Charalambides, **Arya Mazumdar**
Department of CSE and Halıcıoğlu Data Science Institute, University of California, San Diego
Email: ncharalambides@ucsd.edu, arya@ucsd.edu

***Abstract*—In this work, we show that if local datasets in a distributed network are appropriately compressed and then aggregated, it can result in a compressed version of the union of the datasets, in terms of an ℓ_2-subspace embedding. Specifically, we show that *sketching* datasets which are locally generated or stored at a node in a network; via oblivious embeddings, and then aggregated, result in a valid sketch of the collective dataset. The key idea is that by applying distinct random projections on the "*local*" datasets, roughly gives each data point the same importance in the "*global*" dataset. From this, uniform sampling on the local transformed datasets is close to a uniform sampling on the global dataset, after the local projections take place. Our main arguments are also justified numerically.**

I. Introduction and Related Work

Randomization in numerical linear algebra and data science has been a key tool for dimensionality reduction over the past 25 years for handling large datasets [1]–[4], and is an interdisciplinary field of study which is referred to as "*sketching*" and abbreviated to "RandNLA". Sketching offers an effective and cheaper randomized approach to tackling problems such as matrix factorization [5], eigenvalue computation [6], k-means [7], [8], or solving linear systems [9], [10], which are prohibitive in high dimensions and beyond reach for deterministic methods. The core idea behind sketching is to leverage randomization to create structured and well-conditioned matrices that preserve important properties of the original matrices, when compressing them. By applying randomized algorithms to these structured matrices, one can obtain approximate solutions that are statistically close to the true solutions of the original problem.

More recently, with the increase in daily data generation which is prevalent in many machine learning and statistical inference models, resorting to distributed systems for storage and computations is a necessity. A prime example is federated learning, which is also concerned with the privacy and security of data generated locally by users in such systems.

In this article, we show that it is possible to aggregate sketches of local datasets from a distributed network, to obtain a sketch of the collective dataset as a whole (sometimes referred to as the "*global dataset*"). We work with "oblivious subspace embeddings" (OSE); a special case of ℓ_2-subspace embeddings (ℓ_2-s.e.), which suit our objective. Specifically, by first performing a random projection on the local datasets, we show that in the resulting global dataset the transformation of each data point is of approximately equal importance; which is quantified by leverage scores. This then implies that local uniform sampling suffices, as it is close to uniform sampling over the global dataset. Embeddings in such data-distributed settings are useful in variety of data perprocessing/pretraining and other unsupervised learning tasks such as clustering, or PCA [11], [12].

To obtain the sketch of the global dataset, the nodes of the distributed network send their sketches to a coordinator who aggregates them. Depending on the choice of the random projection performed by the nodes, there is a security guarantee on their local information; which prohibits potential eavesdroppers and the coordinator from recovering the data points, which is of increasing importance in distributed machine learning. Ultimately, we obtain a summary of the global dataset in a decentralized manner, without explicitly revealing or aggregating the local data.

The paper is organized as follows. In Sec. II we review notions from RandNLA which we need for our algorithm, as well as introduce notation. In Sec. III we formally present our algorithm and how it applies to distributed settings. The Analysis of our results are then presented in Sec. III-A. Finally, we present numerical experiments in Sec. IV; and concluding remarks in V. All proofs can be found online in [13].

A. Related Work

The closest work to what we present, is a recent sketching technique coined "*block-SRHT*" [14], which considers distributed architectures. This is the only sketching scheme discussed in a recent monograph [15] in which local sketching is performed. In [14], the authors assume that a partition of the global dataset is sent to each of the nodes, who then apply a *Subsampled Randomized Hadamard Transformed* (SRHT) to their allocated partition; along with an additional global permutation and a local signature matrix, before they send back the sketch to the coordinator who sums the sketches.

The main technique used in the block-SRHT is that by applying the global permutation and additional signature matrix, and performing what they call the "sum-reduce" operation, there is resemblance with the standard SRHT [10], [16]. Their overall computational cost is a constant factor higher than sketching with a standard SRHT for the same reduced dimension. On the contrary, the overall computational cost of our approach is a constant factor lower than standard sketches, as we perform multiple smaller sketches and then *aggregate* them, instead of *summing* them. To this extent, our objective is also more general, as we assume the data is already distributed across a network (such as in federated learning), and the global dataset is never itself aggregated. Another drawback of the block-SRHT, is that its overall communication load is a factor of k higher than ours, for the same reduced dimension.

This work was partially supported by NSF Awards 2217058, 2133484.

Other related work include distributed sketching techniques in which different sketches are performed on the global dataset; to solve varying sketched versions of the global system of linear equations, which are then averaged or aggregated to determine a good approximate solution for linear regression [17], [18]. The main drawback here compared to our approach, is that each sketch is performed on the global dataset; and therefore it is not suitable for systems in which local nodes wish not to share their information with other nodes.

II. Preliminaries

In this section, we set up our notation and further describe how our proposed method relates to standard sketching techniques and approaches. One of the most representative applications of RandNLA is the linear least squares approximation problem, in which we seek to approximately solve the overdetermined linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ by solving

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

for $\mathbf{A} \in \mathbb{R}^{N\times d}$ and $\mathbf{b} \in \mathbb{R}^N$, where $N \gg d$. A regularizer $\lambda T(\mathbf{x})$ can also be added to $L_{ls}(\mathbf{A},\mathbf{b};\mathbf{x})$ if desired. In our setting, we assume that $\mathbf{A}$ and $\mathbf{b}$ partitioned across their rows are local datasets $\mathbf{A}_i$ with corresponding labels $\mathbf{b}_i$, *i.e.*

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

where $\mathbf{A}_i \in \mathbb{R}^{n\times d}$ and $\mathbf{b}_i \in \mathbb{R}^n$ for all i, and $n = N/k$; for which $n > d$. We consider the reduced SVD of $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{\mathbb{N}\times d}$ and $\mathbf{A}$ is full rank. To simplify our presentation, we assume that $k|N$; and $\{\mathbf{A}_\iota\}_{\iota=1}^k$ are equipotent. A way to approximate (1) in a faster manner, is to instead solve the modified least squares problem

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

for $\mathbf{S} \in \mathbb{R}^{R\times N}$ an ℓ_2-s.e. sketching matrix, with $R < N$.

Definition 1 (Ch.2 [2])**.** *A sketching matrix $\mathbf{S} \in \mathbb{R}^{R\times N}$ is a ℓ_2-**subspace embedding** of $\mathbf{A} \in \mathbb{R}^{N\times d}$ with a left orthonormal basis $\mathbf{U}$, and $N \gg d$; $N > R > d$, if for any $\mathbf{y} \in \mathsf{im}(\mathbf{U})$ we have with high probability:*

$$(1-\epsilon)\cdot\|\mathbf{y}\| \leqslant \|\mathbf{S}\mathbf{y}\| \leqslant (1+\epsilon)\cdot\|\mathbf{y}\|$$

*for $\epsilon > 0$. This is equivalent to satisfying the ℓ_2-**s.e. property**:*

$$\|\mathbf{I}_d - (\mathbf{S}\mathbf{U})^\top(\mathbf{S}\mathbf{U})\|_2 \leqslant \epsilon \ . \tag{4}$$

In turn, Definition 1 characterizes the approximation's error of the solution $\hat{\mathbf{x}}$ of (3), as

$$\|\mathbf{A}\hat{\mathbf{x}}-\mathbf{b}\|_2 \leqslant (1+\mathcal{O}(\epsilon))\|\mathbf{A}\mathbf{x}^\star-\mathbf{b}\|_2 \tag{5}$$

and $\|\mathbf{A}(\mathbf{x}^\star-\hat{\mathbf{x}})\|_2 \leqslant \epsilon\|(\mathbf{I}_N-\mathbf{U}\mathbf{U}^\top)\mathbf{b}\|_2$ [19]. Furthermore, desired properties of sketching matrices are that they are zero-mean and normalized, *i.e.* $\mathbb{E}[\mathbf{S}] = \mathbf{0}_{R\times N}$ and $\mathbb{E}[\mathbf{S}^\top\mathbf{S}] = \mathbf{I}_N$ [20], which are met by normalized random matrices with i.i.d. standard Gaussian entries.

Notation: We denote $\mathbb{N}_m \coloneqq \{1,2,\ldots,m\}$, and $X_{\{m\}} = \{X_i\}_{i=1}^m$; where X can be replaced by any variable. We consider random square projection matrices; which are represented by $\mathbf{P}$. Sampling matrices are denoted by $\mathbf{\Omega}$, where each row has a single nonzero entry. The uniform sampling distribution $\{1/N,\ldots,1/N\}$ is denoted by $\mathcal{U}_N$, and by $\tilde{\mathcal{U}}_N$ we represent $\mathcal{U}_N$'s approximation through our approach. The index set of $\mathbf{A}$'s rows is denoted by $\mathcal{I}$, *i.e.* $\mathcal{I} = \mathbb{N}_N$. The index set of $\mathbf{A}_i$ is denoted by $\mathcal{I}_i$ for each $i \in \mathbb{N}_k$, *i.e.* $\mathcal{I}_i = \{(i-1)n+1,\ldots,in\}$ and $\mathcal{I} = \bigsqcup_{\iota=1}^k \mathcal{I}_\iota$. Similarly, by $\mathcal{S}$ we denote the index multiset of $\mathbf{A}$'s sampled rows, and $\mathcal{S}_i$ the index multiset of $\mathbf{A}_i$'s sampled rows for each i. By $\mathbf{e}_i$ we denote the i^{th} standard basis vector. The restriction of the identity matrix $\mathbf{I}_N$ to the entries of $\mathcal{I}_\iota$ is represented by $\mathbf{I}_N\big|_{\mathcal{I}_\iota}$. The j^{th} row of matrix $\mathbf{M}$ is represented by $\mathbf{M}_{(j)}$. We use the "$\sim$" (over-script) to denote corresponding quantities of $\mathbf{A}$ in the global sketch $\widehat{\mathbf{A}}$; which we define in Sec. II-A, II-B.

A. Sketching through Leverage Scores

Many sampling algorithms select data points according to the data's *leverage scores* [21], [22]. The leverage scores of $\mathbf{A}$ measure the extent to which the vectors of its orthonormal basis $\mathbf{U}$ are correlated with the standard basis, and define the key structural non-uniformity that must be dealt with when developing fast randomized matrix algorithms; as they characterize the importance of the data points. Leverage scores are defined as $\ell_j \coloneqq \|\mathbf{U}_{(j)}\|_2^2$; and are agnostic to any particular basis, as they are equal to the diagonal entries of the projection matrix $P_{\mathbf{A}} = \mathbf{A}\mathbf{A}^\dagger = \mathbf{U}\mathbf{U}^\top$. The *normalized leverage scores* of $\mathbf{A}$ for each $j \in \mathbb{N}_N$ are

$$\pi_j \coloneqq \left\|\mathbf{U}_{(j)}\right\|_2^2 / \|\mathbf{U}\|_F^2 = \left\|\mathbf{U}_{(j)}\right\|_2^2 / d \ ,$$

and $\pi_{\{N\}}$ form a sampling probability distribution; as $\sum_{\iota=1}^N \pi_\iota = 1$ and $\pi_\iota \geqslant 0$ for all ι. This induced distribution has been proven useful in linear regression [2], [19], [22], [23], as well as a plethora of other applications [3], [7], [24], [25].

The *coherence* of $\mathbf{A}$ is defined as $\gamma \coloneqq \max_{\iota\in\mathbb{N}_N}\{\ell_\iota\}$. Similar to (2), we partition

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1^\top & \cdots & \mathbf{U}_k^\top \end{bmatrix}^\top$$

and define the "*local coherence*" of each data block as $\gamma_i \coloneqq \max_{j\in\mathcal{I}_i}\{\ell_j\}$. We denote the sum of each block's corresponding leverage scores by $\mathscr{L}_i \coloneqq \sum_{j\in\mathcal{I}_i}\ell_j$. It is worth noting that in our setting, the closer $\mathscr{L}_i$ and γ_i are to d/k and d/N respectively, the more homogeneous the local data blocks are; when they considered as a global dataset. If $\gamma_i = d/N$, $\mathbf{U}_i$ is aligned with the standard basis.

Next, we recall the leverage score sampling ℓ_2-s.e. sketch; which is used to obtain $\grave{\mathbf{A}} \coloneqq \grave{\mathbf{S}}\mathbf{A}$. Given $\mathbf{A}$, we sample $R > d$ rows *with replacement* (w.r.) from $\mathbf{A}$ according to $\pi_{\{N\}}$. If at trial j the row i_j was sampled; we rescale it $1/\sqrt{R\pi_{i_j}}$, and set $\grave{\mathbf{A}}_{(j)} = \mathbf{A}_{(i_j)}/\sqrt{R\pi_{i_j}}$. It is clear that here $\grave{\mathbf{S}} \in \mathbb{R}^{R\times N}$ is simply a sampling and rescaling matrix, *i.e.* $\grave{\mathbf{S}}_j = \mathbf{e}_{i_j}^\top/\sqrt{R\pi_{i_j}}$.

In many cases, estimating the leverage scores is preferred, as computing them exactly requires $\mathcal{O}(Nd^2)$ time which is

excessive. We can instead use accurate approximate scores $\hat{\ell}_{\{N\}}$ which can be computed in $\mathcal{O}(Nd\log N)$ time [22]. The estimates are "close" in the following sense: $\hat{\ell}_i \geqslant \beta\ell_i$ for all i, where $\beta \in (0,1]$ is a misestimation factor. The only difference in sampling according to $\hat{\ell}_{\{N\}}$, is that we need to oversample by a factor of $1/\beta$ to get the same theoretical guarantee. The ℓ_2-s.e. result of $\grave{\mathbf{S}}$ is presented next [2], [19], [26].

Theorem 2. *The leverage score sketching matrix $\grave{\mathbf{S}}$ is a ℓ_2-s.e of $\mathbf{A}$. Specifically, for $\delta > 0$ and $R = \Theta\left(d\log\left(2d/\delta\right)/(\beta\epsilon^2)\right)$, the identity of* (4) *is satisfied with probability at least $1-\delta$.*

B. Oblivious Subspace Embeddings

A drawback of directly applying leverage score sampling locally in hope of obtaining a global sketch by aggregating the local sketches, is that the local datasets may be highly heterogeneous, and the sampling performed locally may not be representative of the global leverage score sampling distribution. To alleviate this issue, we resort to OSEs, which exploit random projections and/or uniform sampling. Two prime examples of OSEs, are the Gaussian sketch and the SRHT. It is also worth noting that utilizing random Gaussian and Rademacher random matrices in data compression has close ties to the Johnson-Lindenstrauss lemma [27]–[29], which predates the study of RandNLA.

The Gaussian sketch is defined through a random projection $\mathbf{G} \in \mathbb{R}^{R\times N}$ where $\mathbf{G}_{ij} \sim \mathcal{N}(0,1)$, which is then rescaled to get $\mathbf{S} = \frac{1}{\sqrt{R}}\mathbf{G}$. To unify our techniques, we note that directly applying $\frac{1}{\sqrt{R}}\mathbf{G} \in \mathbb{R}^{R\times N}$ is equivalent to uniformly sampling (without replacement) R rows from a $N\times N$ Gaussian matrix. This is also true for the Rademacher sketch, where $\mathbf{\Theta}_{ij} \sim \mathsf{Unif}(-1,+1)$ and $\mathbf{S} = \frac{1}{\sqrt{R}}\mathbf{\Theta}$. For further speedups with these unstructured projections, one could therefore directly apply $R\times N$ rescaled projections and not consider uniform sampling. A benefit of considering the uniform sampling matrix $\mathbf{\Omega}$ being generated separately to the random projection, is that other sampling matrices may be utilized instead [30], [31].

The SRHT is comprised of three matrices: $\mathbf{\Omega} \in \mathbb{R}^{R\times N}$ a uniform sampling w.r. and rescaling matrix of R rows, $\bar{\mathbf{H}}_N$ the normalized Hadamard matrix of order N:

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

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. If N is not a power of 2, we can pad $\mathbf{A}$ with zeros to meet this requirement. The SRHT sketching matrix is then $\mathbf{S} = \sqrt{\frac{N}{R}} \cdot \mathbf{\Omega}\bar{\mathbf{H}}_N\mathbf{D}$, where $\bar{\mathbf{H}}_N\mathbf{D}$ is unitary matrix that rotates $\mathbf{U}$. The main intuition of the projection is that it expresses the original signal or feature-row in the Walsh-Hadamard basis. Furthermore, $\bar{\mathbf{H}}_N$ can be applied in $\mathcal{O}(Nd\log N)$ time, by using Fourier based methods.

In the new left orthonormal basis of $\mathbf{A}$ after the aforementioned projections are applied, the resulting leverage scores are close to uniform. Hence, uniform sampling is applied through $\mathbf{\Omega}$ to reduce the effective dimension N, whilst the information of $\mathbf{A}$ is maintained. An appropriate rescaling according to the number of sampling trials also takes place, in order to reduce the variance of the resulting estimator.

The idea behind our approach is that the local projections will "flatten" the leverage scores in $\mathbf{A}$ of their local blocks; *i.e.* $\pi_j \approx \mathscr{L}_i/(nd)$ for each $j \in \mathcal{I}_i$ and every $i \in \mathbb{N}_k$. By then locally performing uniform row sampling on $\mathbf{P}_i\mathbf{A}_i$; we get a close to uniform sampling across all the projected blocks.

III. Distributed Local Sketching

In this section, we discuss the details of our distributed sketching scheme. The i^{th} node applies a random projection matrix $\mathbf{P}_i \in \mathbb{R}^{n\times n}$ which is generated locally; in order to flatten the corresponding leverage scores. As we will see, the flattening here is with respect to $\mathbf{U}_i$; *i.e.* w.h.p. $\ell_j \approx \mathscr{L}_i/(nd)$ for each $j \in \mathcal{I}_i$. This is the cost we pay for performing local sketches. Nonetheless, we show in Figure 2 that the flattening degrades gracefully as k increases; even for global datasets with highly non-uniform leverage scores. To partially circumvent this concern if our approach is to be performed by a single user or centrally administered by the coordinator, a random permutation can be applied on the rows of $\mathbf{A}$ before the partitioning takes place.

After locally applying $\mathbf{P}_i$, the nodes randomly sample $r = R/k$ rows from $\mathbf{P}_i\mathbf{A}_i$ which they rescale by $\sqrt{n/r} = \sqrt{N/R}$ and aggregate through $\mathbf{\Omega}_i$; in order to obtain the local sketches

$$\mathbf{S}_i\mathbf{A}_i \in \mathbb{R}^{r\times d}, \quad \text{for} \quad \mathbf{S}_i = \sqrt{n/r}\cdot(\mathbf{\Omega}_i \cdot \mathbf{P}_i) \in \mathbb{R}^{r\times n}.$$

Then, each node communicates the resulting sketch to the coordinator; who aggregates them. The scheme is described algorithmically in 1, and depicted pictorially in Figure 1.

Algorithm 1: Distributed Local Sketching

Input: Effective local dimension r ▷ $R > d$ and $R = rk$
Output: Sketch $\widehat{\mathbf{A}} \in \mathbb{R}^{R\times d}$, of the collective dataset $\mathbf{A}$
for $i = 1$ *to* k **do**
  i^{th} node:
  1) Generate a random $\mathbf{P}_i \in \mathbb{R}^{n\times n}$ ▷ $\mathbb{E}\left[\mathbf{P}_i^\top\mathbf{P}_i\right] = \mathbf{I}_n$
  2) Uniformly sample r rows from $\mathbf{P}_i\mathbf{A}_i$, through $\mathbf{\Omega}_i$
  3) Deliver $\sqrt{\frac{n}{r}}(\mathbf{\Omega}_i\mathbf{P}_i)\,\mathbf{A}_i =: \mathbf{S}_i\mathbf{A}_i$ to the coordinator
  ***note**: 1) and 2) can be performed simultaneously by generating $\mathbf{S}_i \in \mathbb{R}^{r\times n}$, to reduce the local computations*
end
Coordinator: Aggregates $\widehat{\mathbf{A}} = \left[(\mathbf{S}_1\mathbf{A}_1)^\top \ \cdots \ (\mathbf{S}_k\mathbf{A}_k)^\top\right]^\top$

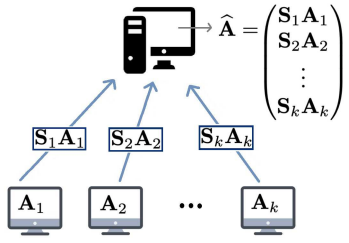

Fig. 1. Schematic of distributed aggregated sketching.

A. Analysis of our approach

For the analysis of our approach, we note that the final sketch $\widehat{\mathbf{A}}$ of $\mathbf{A}$ can be summarized by the "*global sketching*"

matrix $\tilde{\mathbf{S}} \in \mathbb{R}^{R\times N}$, comprised of the "*local sketching*" matrices $\mathbf{S}_{\{k\}}$ across its diagonal:

$$\overbrace{\begin{pmatrix}\mathbf{S}_1 & & \\ & \ddots & \\ & & \mathbf{S}_k\end{pmatrix}}^{\tilde{\mathbf{S}}\in\mathbb{R}^{R\times N}} = \sqrt{\frac{N}{R}} \cdot \overbrace{\begin{pmatrix}\mathbf{\Omega}_1 & & \\ & \ddots & \\ & & \mathbf{\Omega}_k\end{pmatrix}}^{\tilde{\mathbf{\Omega}}\in\{0,1\}^{R\times N}} \cdot \overbrace{\begin{pmatrix}\mathbf{P}_1 & & \\ & \ddots & \\ & & \mathbf{P}_k\end{pmatrix}}^{\tilde{\mathbf{P}}\in\mathbb{R}^{N\times N}}$$

where the sampled index multisets $\mathcal{S}_{\{k\}}$ correspond to the sampling matrices $\mathbf{\Omega}_{\{k\}}$, and $\bigcup_{i=1}^{k}\mathcal{S}_i$ to $\tilde{\mathbf{\Omega}}$.

It is noteworthy that $\tilde{\mathbf{S}}$ can also be interpreted as a sparse sketching matrix, when carried out locally by a single server. By the block diagonal structure of $\tilde{\mathbf{S}}$, the resulting global sketch recovered by the coordinator is:

$$\widehat{\mathbf{A}} \coloneqq \tilde{\mathbf{S}}\mathbf{A} = \left[(\mathbf{S}_1\mathbf{A}_1)^\top \ \cdots \ (\mathbf{S}_k\mathbf{A}_k)^\top\right]^\top \in \mathbb{R}^{R\times d}.$$

Note that $\sqrt{R/N}$ and $\tilde{\mathbf{\Omega}}$ commute. Since $\sqrt{R/N}\cdot\tilde{\mathbf{P}}$ is a block diagonal matrix, the corresponding blocks of $\mathbf{U}$ are rotated/transformed by their respective projections, *i.e.*:

$$\tilde{\mathbf{U}} \coloneqq \left(\sqrt{N/R}\cdot\tilde{\mathbf{P}}\right)\cdot\mathbf{U} = \left[\tilde{\mathbf{U}}_1^\top \ \cdots \ \tilde{\mathbf{U}}_k^\top\right]^\top$$

where $\tilde{\mathbf{U}}_i = \left(\sqrt{n/r}\cdot\mathbf{P}_i\right)\cdot\mathbf{U}_i$ for each $i\in\mathbb{N}_k$. We denote the leverage scores of $\tilde{\mathbf{P}}\cdot\mathbf{A}$ by $\tilde{\ell}_{\{N\}}$.

Next, we show that the leverage scores of each $\tilde{\mathbf{U}}_i$ are flattened with respect to $\mathscr{L}_i$ by the local projection. For our analysis, we assume that $\mathbf{P}_i$ is a random unitary matrix, drawn from an arbitrary large finite subset $\tilde{O}_n(\mathbb{R})$ of the set orthonormal matrices $O_n(\mathbb{R})$ of size $n\times n$. In practice, normalized Gaussian and Rademacher matrices are used; as they satisfy $\mathbb{E}\left[\mathbf{G}^\top\mathbf{G}\right] = \mathbf{I}_N$. The other option which is widely used is the unitary Randomized Hadamard Transform. Analogous results can also be derived for these options of $\mathbf{P}_i$.

It is important to note that $O_n(\mathbb{R})$ is a regular submanifold of the general linear group $\mathrm{GL}_n(\mathbb{R})$. Hence, we can define a distribution on any subset of $O_n(\mathbb{R})$. For simplicity, we consider the uniform distribution. A simple method of generating a random matrix that follows the uniform distribution on the Stiefel manifold $V_m(\mathbb{R}^m)$ can be found in [32, Theorem 2.2.1]. Alternatively, one could generate a random Gaussian matrix and then perform Gram–Schmidt to orthonormalize it.

Lemma 3. *Consider a fixed $i\in\mathbb{N}_k$. Assume that $\mathbf{P}_i$ is arbitrarily drawn from $\tilde{O}_n(\mathbb{R})$. Then, for any $j\in\mathcal{I}_i$, we have $\mathbb{E}[\tilde{\ell}_j] = \mathscr{L}_i/n$.*

Proposition 4. *For a fixed $i\in\mathbb{N}_k$ and $\xi>0$, the normalized (w.r.t. $\tilde{\mathbf{U}}$) leverage scores $\{\bar{\ell}_j\}_{j\in\mathcal{I}_i}$ corresponding to $\tilde{\mathbf{U}}_i$ satisfy*

$$\Pr\left[|\bar{\ell}_j - \mathscr{L}_i/(nd)| < \zeta\right] \geqslant 1-\xi$$

for any $\zeta\geqslant\zeta' \coloneqq \frac{\mathscr{L}_i}{d}\sqrt{\log(2/\xi)/2}$.

In Figure 2, we show numerically the flattening of the normalized leverage scores of a random $\mathbf{A}\in\mathbb{R}^{2000\times 40}$ following a t-distribution, which scores were highly non-uniform. In this experiments, $\mathbf{P}_{\{k\}}$ were random Gaussian matrices. As noted previously, random Gaussian matrices are good surrogates for unitary random matrices, and are widely used in practice as they are approximately orthogonal. Furthermore, we observe that the flattening degrades gracefully as k increases, and if a random permutation on $\mathbf{A}$'s rows is applied before $\tilde{\mathbf{P}}$, we have slightly better results for each k. Analogous simulation results were observed when the $\mathbf{P}_i$'s were randomized Hadamard transforms, random unitary, and Rademacher random matrices.

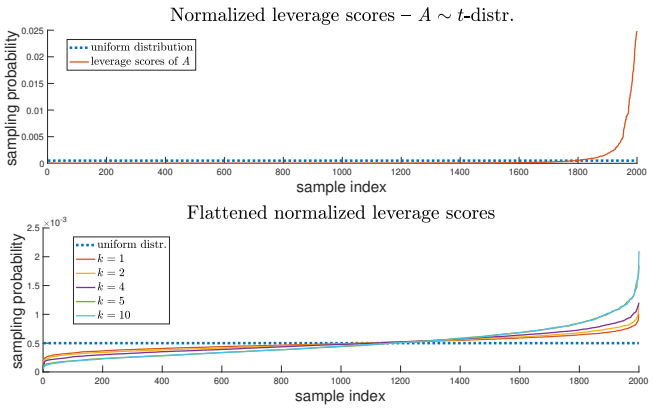

Fig. 2. Flattening of leverage scores distribution, for $\mathbf{A}\sim t$-distribution.

By Proposition 4, the quality of the flattening approximations of the local blocks depends on the sum of the local leverage scores $\mathscr{L}_i$. To give better theoretical guarantees for our approach and analysis, we could make the assumption that $\mathscr{L}_i \approx d/k$ for each i. Such an assumption is weaker to analogous assumptions in distributed sketching algorithms which assume that $\mathbf{A}$ has a low coherence, *e.g.* [33]. This is akin to assuming that $\gamma_i \approx d/N$ for each $i\in\mathbb{N}_k$. We believe that it is not possible to improve on the local flattening algorithmically without exchanging information or aggregating the data a priori, as the objective is to flatten the scores of the collective $\mathbf{A}$. Further investigating this is worthwhile future work. We alleviate this concern by oversampling according to an appropriate misestimation factor $\tilde{\beta}$.

Before we present our main result, we need to show that $\mathbf{\Omega}$ is close to being a uniform sampling (w.r.) matrix of R out of N rows. In Proposition 5, we show this by applying Chebyshev's inequality to the balls into bins problem. Specifically, it shows that w.h.p. the sampling of $\mathcal{S}$ in $\mathcal{I}$ is close to a sampling of $\bigcup_{i=1}^{k}\mathcal{S}_i$, *i.e.* the cardinality of sampled indices of $\mathcal{S}$ that lie in any $\mathcal{I}_i$ is not far from $\#\mathcal{S}_i = r$. The important factors to note are that all sampling trials in both scenarios are uniform, identical, independent and with replacement.

Proposition 5. *Partition the sampled index set $\mathcal{S}$ into ordered partitions S_i of $\mathcal{I}$; according to $\mathcal{I}_{\{k\}}$, i.e. $S_i = \mathcal{S}\bigcap\left(\bigcup_{l=1}^{R}\mathcal{I}_i\right)$. Then, for any $i\in\mathbb{N}_k$: $\Pr\left[|\#S_i - r|\geqslant 10\right] \leqslant 1/100$.*

Remark 6. *In order to get an exact global sampling index set for $\mathcal{I}$, the coordinator could determine $\mathcal{S}$ locally and then request the nodes to send their respective projected rows, or a subset of corresponding cardinality. In essence, this is similar to the "unique sampling matrix $\mathbf{R}$" of the block-SRHT [14].*

Next, we provide our main result regarding the ℓ_2-s.e. of the aggregated dataset $\mathbf{A}$, through local sketching.

Theorem 7. *Let $\mathbf{P}_{\{k\}}$ of Algorithm 1 be random unitary matrices, and $\tilde{\beta} = \frac{k}{d} \cdot \min_{i\in\mathbb{N}_k} \{\mathscr{L}_i\}$. Then, for $\delta > 0$ and $R = \Theta\left(d \log\left(2d/\delta\right)/(\tilde{\beta}\epsilon^2)\right)$, the sketching matrix $\tilde{\mathbf{S}}$ of the global $\mathbf{A}$, satisfies* (4) *with high probability.*

Corollary 8. *Consider $\frac{1}{\sqrt{r}}\mathbf{G}_{\{k\}}$ rescaled random Gaussian matrices, and perform Gram-Schmidt to each projection to obtain $\hat{\mathbf{P}}_{\{k\}}$. Then, for $\delta > 0$ and $R = \Theta\left(d \log\left(2d/\delta\right)/(\tilde{\beta}\epsilon^2)\right)$, the sketching matrix $\tilde{\mathbf{S}}$ of the global $\mathbf{A}$, w.h.p. satisfies* (4).

We point out that the failure probability of Theorem 7 is higher than that of Theorem 2, as there is also a source of error from Proposition 5 and the flattening of the leverage scores. Therefore, the higher k is, the greater the failure probability is. Experimentally though, we observe that the increase in error is not drastic. Furthermore, if we were to assume that $\mathscr{L}_i \approx d/k$, *i.e.* the local datasets are homogeneous with respect to $\mathbf{U}$, then the misestimation factor $\tilde{\beta}$ would be close to 1. In the general setting we are considering, it is best to avoid such assumptions in practice.

It would be interesting to investigate whether one can estimate $\mathscr{L}_{\{k\}}$, without directly aggregating or sharing the data. If so, the flattening of the scores through the local sketches could potentially be improved.

Furthermore, the approach of Corollary 8 suggests that each node performs a Gram-Schmidt process on its generated random matrix. The benefits of this is for the analysis of our technique, and can be avoided in practice, as these matrices satisfy $\mathbb{E}[\mathbf{S}^\top\mathbf{S}] = \mathbf{I}_N$.

Another major benefit of considering random Gaussian matrices is their security aspect. Under the assumption that $\mathbf{A}_i$ is randomly sampled from a distribution with finite variance, then the mutual information per symbol between $\mathbf{S}_i\mathbf{A}_i$ (sketch of $\mathbf{A}_i$ observed by the coordinator) and $\mathbf{A}_i$, has a logarithmic upper bound in terms of the variance; which approaches zero as n increases or if r is selected appropriately, *e.g.* [18], [34]–[36]. This implies information-theoretic security and privacy of the local data blocks. Similar results were obtained in [37], [38] for $\mathbf{P}_{\{k\}}$ random unitary matrices, which made different assumptions on $\mathbf{A}_i$ and the distribution of $\mathbf{U}_i$.

Proposition 9. *Assume that $\mathbf{A}_i$ is drawn from a distribution with finite variance. Then, the rate at which information about $\mathbf{A}_i$ is revealed by the compressed data $\mathbf{S}_i\mathbf{A}_i$ for $\mathbf{S}_i \in \mathbb{R}^{r\times n}$ a Gaussian sketch, satisfies $\sup \frac{I(\mathbf{A}_i;\mathbf{S}_i\mathbf{A}_i)}{nd} = \mathcal{O}(r/n) \to 0$. Specifically, the original $\mathbf{A}_i$ and the observed $\mathbf{S}_i\mathbf{A}_i$ are statistically independent, which means we obtain perfect secrecy for a small enough sketch dimension r.*

IV. Experiments

In the following experiments we considered the errors according to the ℓ_2-s.e. error (4) and relative regression error (5), for $\mathbf{A} \in \mathbb{R}^{18000\times 40}$ following a t-distribution, similar to the experiment of Figure 2. We considered R varying from 30% to 90% of N, for different values of k, and $\mathbf{P}_i$ rescaled Gaussian matrices. Through these experiments, we convey that the difference in error is small, while we save on computing on the transformed blocks $\{\mathbf{P}_i\mathbf{A}_i\}_{i=1}^k$ by a factor of $(k_1/k_2)^2$ on the overall computation when we move from $k = k_1$ to a smaller $k = k_2$. By combining steps 1) and 2) of Algorithm 1, we require only $\mathcal{O}(rndk)$ operations by each server. Moreover, more accurate approximations were observed with higher N.

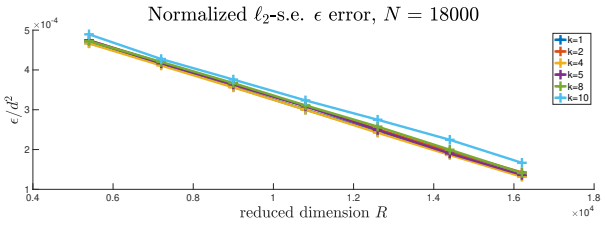

Fig. 3. Normalized ℓ_2-s.e. error (4), for varying k and R.

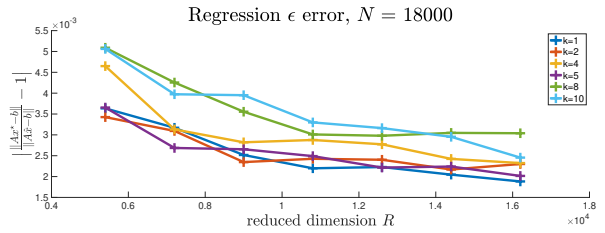

Fig. 4. Regression error (5), for varying k and R.

V. Future Directions and Concluding Remarks

In this work, we showed how we can obtain a global sketch of data distributed across a network in terms of a spectral guarantee, by performing local sketches and not aggregating until after the sketchings have taken place. We also discussed the privacy aspect of this approach, as the local sketchings can provide security guarantees. A potential future direction was mentioned in Sec. III-A, in terms of improving the flattening of the leverage scores distributively. Another interesting avenue is investigating the ideas presented in this paper; when used for sparser/distributed Johnson-Lindenstrauss transforms, with partitions across the features of the data points. Finally, there are potential methods which can tie this approach to that of [14], while still considering our decentralized setting, which can lead to further insights in distributed sketching.

Moreover, our approach was motivated by federated learning, where data is only available to local servers who may wish not to reveal their information. Our proposed approach can be utilized in first order federated algorithms to further accelerate them. It also permits exchange of local sketches for better approximations by correlating the sketches, while also providing a global summary of the data if desired. Moreover, by obtaining a compressed summary of the global dataset, one can administer global updates of mainstream federated approaches through distributed or centralized first and second order methods, while still meeting the objective of keeping the local data secret. Specific applications include PCA and low-rank recovery [39], [40], and subspace tracking [41]. Further investigation of these connections is worthwhile future work.

REFERENCES

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

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

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

[4] ——, "Lectures on Randomized Numerical Linear Algebra," *arXiv preprint arXiv:1712.08880*, 2017.

[5] P.-G. Martinsson, V. Rokhlin, and M. Tygert, "A randomized algorithm for the decomposition of matrices," *Applied and Computational Harmonic Analysis*, vol. 30, no. 1, pp. 47–68, 2011.

[6] W. Swartworth and D. P. Woodruff, "Optimal Eigenvalue Approximation via Sketching," in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, 2023, pp. 145–155.

[7] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized Dimensionality Reduction for k-means Clustering," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 1045–1062, 2014.

[8] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu, "Dimensionality Reduction for k-Means Clustering and Low Rank Approximation," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 2015, pp. 163–172.

[9] T. Sarlós, "Improved Approximation Algorithms for Large Matrices via Random Projections," in *2006 47th annual IEEE symposium on foundations of computer science (FOCS '06)*. IEEE, 2006, pp. 143–152.

[10] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, "Faster Least Squares Approximation," *Numerische mathematik*, vol. 117, no. 2, pp. 219–249, 2011.

[11] S. Guha and N. Mishra, "Clustering Data Streams," in *Data stream management: processing high-speed data streams*. Springer, 2016, pp. 169–187.

[12] V. Gandikota, A. Mazumdar, and A. S. Rawat, "Reliable Distributed Clustering with Redundant Data Assignment," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 2556–2561.

[13] N. Charalambides and A. Mazumdar, "Distributed Local Sketching for ℓ_2 Embeddings," 2024. [Online]. Available: https://drive.google.com/file/d/1Yteaxoq-zK2FCSGZsyGkkFwCEimwKcwU/view?usp=sharing

[14] O. Balabanov, M. Beaupère, L. Grigori, and V. Lederer, "Block Subsampled Randomized Hadamard Transform for Nyström Approximation on Distributed Architectures," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 1564–1576.

[15] R. Murray, J. Demmel, M. W. Mahoney, N. B. Erichson, M. Melnichenko, O. A. Malik, L. Grigori, P. Luszczek, M. Dereziński, M. E. Lopes *et al.*, "Randomized Numerical Linear Algebra: A Perspective on the Field With an Eye to Software," *arXiv preprint arXiv:2302.11474*, 2023.

[16] 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.

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

[18] B. Bartan and M. Pilanci, "Distributed Sketching for Randomized Optimization: Exact Characterization, Concentration and Lower Bounds," *IEEE Transactions on Information Theory*, 2023.

[19] A. Eshragh, F. Roosta, A. Nazari, and M. W. Mahoney, "LSAR: Efficient Leverage Score Sampling Algorithm for the Analysis of Big Time Series Data," *Journal of Machine Learning Research*, vol. 23, no. 22, pp. 1–36, 2022. [Online]. Available: http://jmlr.org/papers/v23/20-247.html

[20] M. R. Rodrigues and Y. C. Eldar, *Information-Theoretic Methods in Data Science*. Cambridge University Press, 2021.

[21] P. Ma, M. W. Mahoney, and B. Yu, "A Statistical Perspective on Algorithmic Leveraging," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 861–911, 2015.

[22] 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.

[23] M. W. Mahoney, "Lecture Notes on Randomized Linear Algebra," *arXiv preprint arXiv:1608.04481*, 2016.

[24] D. A. Spielman and N. Srivastava, "Graph Sparsification by Effective Resistances," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, 2011.

[25] B. Ordozgoiti, A. Matakos, and A. Gionis, "Generalized Leverage Scores: Geometric Interpretation and Applications," in *International Conference on Machine Learning*. PMLR, 2022, pp. 17 056–17 070.

[26] N. Charalambides, M. Pilanci, and A. O. Hero III, "Gradient Coding through Iterative Block Leverage Score Sampling," *arXiv preprint arXiv:2308.03096*, 2023.

[27] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," in *Contemp. Math.*, vol. 26, 1984, pp. 189–206.

[28] D. Achlioptas, "Database-friendly Random Projections: Johnson-Lindenstrauss with binary coins," *Journal of computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[29] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of Johnson and Lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[30] J. Nelson and H. L. Nguyên, "OSNAP: Faster Numerical Linear Algebra Algorithms via Sparser Subspace Embeddings," in *2013 ieee 54th annual symposium on foundations of computer science*. IEEE, 2013, pp. 117–126.

[31] M. Derezinski, J. Lacotte, M. Pilanci, and M. W. Mahoney, "Newton-LESS: Sparsification without trade-offs for the sketched Newton update," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2835–2847, 2021.

[32] Y. Chikuse, *Statistics on Special Manifolds*, ser. Lecture Notes in Statistics. Springer New York, 2012. [Online]. Available: https://books.google.com.cy/books?id=7lX1BwAAQBAJ

[33] S. Wang, F. Roosta, P. Xu, and M. W. Mahoney, "GIANT: Globally Improved Approximate Newton Method for Distributed Optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

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

[35] S. Zhou, J. Lafferty, and L. Wasserman, "Compressed and Privacy-Sensitive Sparse Regression," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 846–866, 2009.

[36] M. Bloch, O. Günlü, A. Yener, F. Oggier, H. V. Poor, L. Sankar, and R. F. Schaefer, "An Overview of Information-Theoretic Security and Privacy: Metrics, Limits and Applications," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 5–22, 2021.

[37] N. Charalambides, H. Mahdavifar, M. Pilanci, and A. O. Hero III, "Orthonormal Sketches for Secure Coded Regression," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 826–831.

[38] N. Charalambides, H. Mahdavifar, M. Pilanci, and A. O. Hero, "Iterative Sketching for Secure Coded Regression," *IEEE Journal on Selected Areas in Information Theory*, 2024.

[39] A. Grammenos, R. Mendoza Smith, J. Crowcroft, and C. Mascolo, "Federated Principal Component Analysis," *Advances in neural information processing systems*, vol. 33, pp. 6453–6464, 2020.

[40] A. P. Singh and N. Vaswani, "Byzantine-Resilient Federated PCA and Low Rank Matrix Recovery," *arXiv preprint arXiv:2309.14512*, 2023.

[41] P. Narayanamurthy, N. Vaswani, and A. Ramamoorthy, "Federated Over-Air Subspace Tracking From Incomplete and Corrupted Data," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3906–3920, 2022.

APPENDIX A
PROOFS AND FURTHER REMARKS

In this Appendix, we include the missing proofs of the main manuscript.

Proof. [Lemma 3] Recall that for any $j \in \mathcal{I}_i$:

$$\tilde{\ell}_j = \|\tilde{\mathbf{U}}_{(j)}\|_2^2 = \|\mathbf{e}_i^\top \tilde{\mathbf{U}}\|_2^2 = \mathbf{e}_j^\top \tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top \mathbf{e}_j \ .$$

It then follows that

$$\begin{aligned}
\mathbb{E}[\tilde{\ell}_j] &= \mathbb{E}\left[\mathrm{tr}(\mathbf{e}_j^\top \tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top \mathbf{e}_j)\right] \\
&= \mathbb{E}\left[\mathrm{tr}(\mathbf{e}_j\mathbf{e}_j^\top \cdot \tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top)\right] \\
&\overset{\sharp}{=} \sum_{l\in\mathcal{I}_i} \frac{1}{|\mathcal{I}_i|} \cdot \mathrm{tr}\left(\mathbf{e}_l\mathbf{e}_l^\top \cdot \tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top\right) \\
&= \frac{1}{n} \cdot \mathrm{tr}\left(\sum_{l\in\mathcal{I}_i} \mathbf{e}_l\mathbf{e}_l^\top \cdot \tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top\right) \\
&\overset{\flat}{=} \frac{1}{n} \cdot \mathrm{tr}\left(\mathbf{I}_N\big|_{\mathcal{I}_i} \cdot \tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top\right) \\
&= \frac{1}{n} \cdot \mathrm{tr}\left(\tilde{\mathbf{U}}_i\tilde{\mathbf{U}}_i^\top\right) \\
&= \frac{\mathscr{L}_i}{n} \ .
\end{aligned}$$

In $\sharp$ we invoked the definition of expectation, to express $\mathbb{E}[\tilde{\ell}_j]$ in terms of the leverage scores of the transformed block $\tilde{\mathbf{U}}_i$. Further note that in $\flat$, the matrix $\mathbf{I}_N\big|_{\mathcal{I}_i}$ acts similar to a sampling matrix on the rows of $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top$, as when we multiply with $\mathbf{I}_N\big|_{\mathcal{I}_i}$ we only retain the rows indexed by $\mathcal{I}_i$ of $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top$; while the remaining rows are set to zero. □

What we did not present in this work, is that when we consider other random projection matrices $\mathbf{P}_{\{k\}}$; *e.g.* normalized Gaussian or Rademacher random matrices, similar results to Lemma 3 hold. This was observed experimentally, and has been justified for some cases in [38]. Showing this for other random projections is worthwhile future work.

Proof. [Proposition 4] We know that $\tilde{\ell}_j \in [0, \mathscr{L}_i]$, and the normalized scores are $\bar{\ell}_j = \tilde{\ell}_j/d$ for each $j \in \mathcal{I}_i$. By Lemma 3, it follows that

$$\mathbb{E}[\bar{\ell}_j] = \mathbb{E}[\tilde{\ell}_j/d] = \frac{1}{d} \cdot \mathbb{E}[\tilde{\ell}_i] = \frac{\mathscr{L}_i}{nd}.$$

By applying Hoeffding's Inequality [23] for a fixed $\zeta \geqslant \zeta'$:

$$\Pr\left[|\bar{\ell}_j - \mathscr{L}_i/(nd)| < \zeta\right] > 1 - 2e^{-2(\zeta d/\mathscr{L}_i)^2} \geqslant 1 - \xi \ .$$

□

Proof. [Proposition 5] We first note that $\bigcup_{l=1}^R \mathcal{I}_i$ is the multi-set union of R copies of the index set $\mathcal{I}_i$, as we are considering sampling with replacement. In our context, $\mathcal{I}_{\{k\}}$ represent the bins, and the allocation of the R balls into the bins are represented by $\mathcal{S}$. The sub-multiset S_i indicates which balls fell into the i^{th} bin. Let B_{ji} be the indicator random variable

$$B_{ji} = \begin{cases} 1 & \text{if ball } j \text{ falls into bin } i \\ 0 & \text{otherwise} \end{cases}$$

for which $\Pr[B_{ji}] = 1/k$ and $\mathbb{E}[B_{ji}] = 1/k$, as B_{ji} follows a Bernoulli distribution. Further define $Y^i \coloneqq \sum_{j=1}^R B_{ji}$; which follows a Binomial distribution, hence

$$\mathbb{E}\left[Y^i\right] = R/k = r \quad \text{and} \quad \mathrm{Var}\left[Y^i\right] = (R-1)/R \ .$$

It is clear that $Y^i = \#S_i$, so by Chebyshev's inequality the proof is complete. □

Proof. [Theorem 7] For each $i \in \mathbb{N}_k$ and every $j \in \mathcal{I}_i$, by Proposition 4 we can assume that $\bar{\ell}_j \approx \mathscr{L}_i/(nd)$ (w.h.p.), where $\bar{\ell}_j$ is the sampling probability $\tilde{\pi}_j$ that row j is sampled at each independent trial. Hence, Algorithm 1 is now performing approximate leverage score sampling with distribution $\tilde{\mathcal{U}}_N = \bigcup_{\iota=1}^k \left(\bigcup_{j\in\mathcal{I}_\iota} \{\mathscr{L}_\iota/(nd)\}\right)$. The misestimation factor for sampling according to $\mathcal{U}_N$ instead of $\tilde{\mathcal{U}}_N$, is

$$\tilde{\beta} = \min_{\iota\in\mathbb{N}_k} \left\{\frac{\mathscr{L}_\iota/(nd)}{1/N}\right\} = \min_{\iota\in\mathbb{N}_k} \left\{\mathscr{L}_\iota \cdot (k/d)\right\} \ .$$

By Proposition 5, the resulting $\tilde{\mathbf{\Omega}}$ through the local sampling matrices $\mathbf{\Omega}_{\{k\}}$, is close to a uniform sampling matrix (w.r.) of R out of N elements. By applying the above conclusions to Theorem 2, the proof is complete. □

Proof. [Corollary 8] Fix an $i \in \mathbb{N}_k$. By the Gram-Schmidt procedure on $\frac{1}{\sqrt{r}}\mathbf{G}_i$, the resulting $\hat{\mathbf{P}}_i$ is a random unitary matrix drawn from $O_n(\mathbb{R})$. This holds true for all $i \in \mathbb{N}_k$. The claim then follows directly from Theorem 7. □

Proof. [Proposition 9] This follows directly from Theorems 5.1 and 5.2 of [34] when applied to our setting, and the definition of perfect secrecy [36]. □

One thing that was not discussed in the main body of this paper, is the security guaranteed by applying a random projection drawn from $O_n(\mathbb{R})$. To give information-theoretic security guarantees, one needs to make some mild but necessary assumptions regarding the sketching on algorithm which applies a random projection and then samples to construct $\mathbf{S}_i$, and the data matrix $\mathbf{A}_i$ [38]. For a fixed $i \in \mathbb{N}_k$, the message space $\mathcal{M}_i$ needs to be finite, which $\mathcal{M}_i$ in our case corresponds to the set of possible orthonormal bases of the column-space of $\mathbf{A}_i$. 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 $\mathcal{M}_i$ is finite. For this reason, we consider a finite multiplicative subgroup $(\tilde{O}_{\mathbf{A}_i}, \cdot)$ of $O_n(\mathbb{R})$ (thus $\mathbf{I}_n \in \tilde{O}_{\mathbf{A}_i}$, and if $\mathbf{Q} \in \tilde{O}_{\mathbf{A}_i}$ then $\mathbf{Q}^\top \in \tilde{O}_{\mathbf{A}}$), which contains all potential orthonormal bases of $\mathbf{A}_i$.

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 $\mathcal{M}_i = \tilde{O}_{\mathbf{A}_i}$, and assume $\mathbf{U}_{\mathbf{A}_i}$ the $n \times n$ orthonormal basis of $\mathbf{A}_i$ is drawn from $\mathcal{M}_i$ with respect to a distribution D. For simplicity, 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 [32, Theorem 2.2.1]. Alternatively,

one could generate a random Gaussian matrix and then perform Gram–Schmidt to orthonormalize it.

Furthermore, this approach resembles the *one-time pad*, which is one of the few encryption schemes known to provide perfect secrecy. The main difference between the two approaches, is that the spaces we work over are the multiplicative group $(\tilde{O}_{\mathbf{A}_i}, \cdot)$ whose identity is $\mathbf{I}_n$ in, and the additive group $\big((\mathbb{Z}_2)^m, +\big)$ in the one-time pad; whose identity is the zero vector of length m.