The problem of ℓ_2-s.e.; a form of spectral approximation of a matrix, has been extensively studied in RandNLA. The main techniques for constructing appropriate sketching ℓ_2-s.e. matrices, are performing a random projection or row-sampling. Well-known choices of $\mathbf{S}$ for reducing the effective dimension N to r include: i) *Gaussian projection*; for a matrix $\mathbf{\Theta} \sim \mathcal{N}(0,1)$ define $\mathbf{S} = \frac{1}{\sqrt{r}}\mathbf{\Theta}$, ii) *leverage score sampling*; sample with replacement r rows from the matrix according to its normalized leverage score distribution and rescale them appropriately, iii) *Subsampled Hadamard Transform* (SRHT); apply a Hadamard transform and a random signature matrix to judiciously make the leverage scores approximately uniform and then follow similar steps to the leverage score sampling procedure.

In this paper, we first generalize ii) to appropriately sample *submatrices* instead of rows to attain a ℓ_2-s.e guarantee. We refer to such approaches as *block sampling*. Throughout this paper, sampling is done with replacement (w.r.). Sampling blocks has been explored in "block-iterative methods" for solving systems of linear equations [**?**], [**?**], [**?**], [**?**]. Our motivation in dealing with blocks rather than individual vectors, is the availability to invoke results that can be used to characterize the approximations of distributed computing networks, to speed up first-order methods, as sampling individual rows/columns is prohibitive in real-world environments. This in turn leads to an *iterative sketching* approach, which has been well studied in terms of second-order methods [**?**], [**?**], [**?**]. By iterative sketching, we refer to an iterative algorithm which uses a new sketch $\mathbf{S}_{[s]}$ at each iteration. The scenario where a single sketch $\mathbf{S}$ is applied before the iterative process, is referred to as the "sketch-and-solve paradigm" [**?**].

Second, we propose a general framework which incorporates our sketching algorithm into a CC approach. This framework accommodates a central class of sketching algorithms, that of importance (block) sampling algorithms (*e.g.* CUR decomposition [**?**], CR-multiplication [**?**]). Coded computing is a novel computing paradigm that utilizes coding theory to effectively inject and leverage data and computation redundancy to mitigate errors and slow or non-responsive servers; known as *stragglers*, among other fundamental bottlenecks, in large-scale distributed computing. In our setting, the straggling effect is due to computations being communicated over *erasure channels*, whose erasure probability follows a probability distribution which is central to the CC probabilistic model. The seminal work of [**?**] which first introduced CC, focused on exact matrix-vector multiplication and data shuffling. More recent works deal with recovering good approximations, while some have utilized techniques from RandNLA; *e.g.* [**?**], [**?**], [**?**], [**?**], [**?**], [**?**]. Our results are presented in terms of the standard CC probabilistic model proposed in [**?**], though they extend to any computing network comprised of a central server and computing nodes, referred to as *servers*.

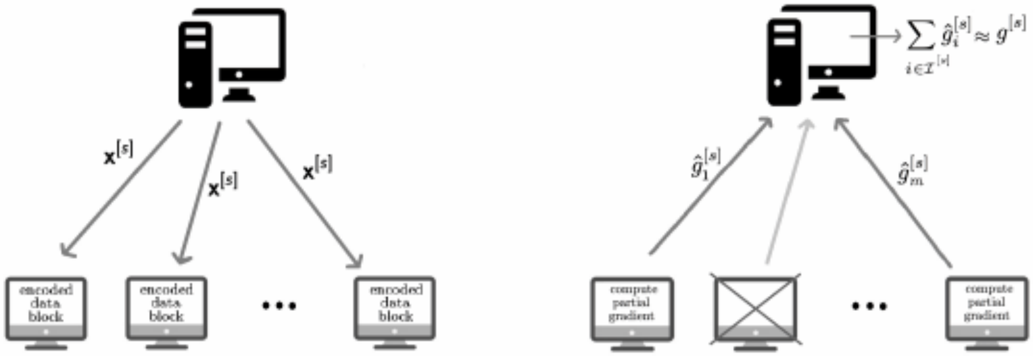

Fig. 1: Schematic of our approximate GC scheme, at iteration s. Each server has an encoded block of data, of which they compute the gradient once they receive the updated parameters $\mathbf{x}^{[s]}$. The central server then aggregates a subset of all the gradients $\{\hat{g}_j^{[s]}\}_{j=1}^m$, indexed by $\mathcal{I}^{[s]}$, to approximate the gradient $g^{[s]}$. At each iteration we expect a different index set $\mathcal{I}^{[s]}$, which leads to iterative sketching.

To mitigate stragglers, we appropriately encode and replicate the data blocks, which leads to accurate CC estimates. In contrast to previous works which simply replicate each computational task or data block the same number of times [**?**], [**?**], [**?**], [**?**], we replicate blocks according to their *block leverage scores*. Consequently, this induces a non-uniform sampling distribution in the aforementioned CC model; which is an approximation to the normalized block leverage scores. A drawback of using RandNLA techniques is that exact computations are not recoverable, though our method does not require a decoding step, a task of high complexity and a prevalent bottleneck in CC. For more details on the various directions of CC, the reader is referred to the monographs [**?**], [**?**].

The central idea of our approach is that non-uniform importance sampling can be emulated, by replicating tasks across the network's servers, who communicate through an erasure channel. The tasks' computation times follow a runtime distribution [**?**], which along with a prespecified gradient transmission "*ending time*" T, determine the number of replications per task across the network. Though similar ideas have been proposed [**?**], [**?**], [**?**], [**?**]; where sketching has been incorporated into CC, this is the first time redundancy is introduced through RandNLA; as opposed to compression, to obtain approximation guarantees. In terms of CC, though uniform replication is a very powerful technique, it does not capture the relevance between the information of the dataset. We capture such information through replication and rescaling according to the block leverage scores. By then allowing *uniform* sampling of these blocks, we attain a spectral approximation. In the CC setting this then corresponds to an iterative

Sketching matrices are represented by $\mathbf{S}$ and $\widetilde{\mathbf{S}}_{[s]}$. The script $[s]$ indexes an iteration $s = 0, 1, 2, \dots$ which we drop when it is clear from the context. We will be reducing dimension N to r, *i.e.* $\mathbf{S} \in \mathbb{R}^{r \times N}$. Sampling matrices are denoted by $\mathbf{\Omega} \in \{0,1\}^{r \times N}$, and diagonal rescaling matrices by $\mathbf{D} \in \mathbb{R}^{N \times N}$.

Approximate block sampling distributions to $\Pi_{\{K\}}$ are denoted by $\tilde{\Pi}_{\{K\}}$, and the distributions induced through expansion networks by $\bar{\Pi}_{\{K\}}$. We quantify the difference between distributions $\Pi_{\{K\}}$ and $\tilde{\Pi}_{\{K\}}$ by the distortion metric $d_{\Pi,\tilde{\Pi}} := \frac{1}{K} \sum_{i=1}^{K} |\Pi_i - \tilde{\Pi}_i|$, which is the ℓ_1 distortion between $\Pi_{\{K\}}$ and $\tilde{\Pi}_{\{K\}}$, *e.g.* [**?**].

A. *Least Squares Approximation*

Least squares approximation is a technique to find an approximate solution to a system of linear equations that has no exact solution, and has found applications in many fields [**?**]. Consider the system $\mathbf{Ax} = \mathbf{b}$, for which we want to find an approximation to the best-fitted

$$\mathbf{x}^\star = \underset{\mathbf{x} \in \mathbb{R}^d}{\arg\min} \Big\{ L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}) := \|\mathbf{Ax} - \mathbf{b}\|_2^2 \Big\}, \tag{2}$$

which objective function L_{ls} has gradient

$$g^{[s]} = \nabla_{\mathbf{x}} L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[s]}) = 2\mathbf{A}^\top(\mathbf{Ax}^{[s]} - \mathbf{b}). \tag{3}$$

We refer to the gradient of the block pair $(\mathbf{A}_i, \mathbf{b}_i)$ from (**??**) as the i^{th} *partial gradient*; $g_i^{[s]} = \nabla_{\mathbf{x}} L_{ls}(\mathbf{A}_i, \mathbf{b}_i; \mathbf{x}^{[s]})$. Existing exact methods find a solution vector $\mathbf{x}^\star$ in $O(Nd^2)$ time, where $\mathbf{x}^\star = \mathbf{A}^\dagger \mathbf{b}$. In Subsection **??** we focus on approximating the optimal solution $\mathbf{x}^\star$ by using our methods, via distributive SD/SSD and iterative sketching. What we present also accommodates regularizers of the form $\lambda \|\mathbf{x}\|_2^2$, though to simplify our expressions, we only consider (**??**).

B. *Steepest Descent*

When considering a minimization problem with a convex differentiable objective function $L \colon \mathbb{R}^d \to \mathbb{R}$, we select an initial $\mathbf{x}^{[0]} \in \mathbb{R}^d$ and repeat at iteration $s+1$: $\mathbf{x}^{[s+1]} \leftarrow \mathbf{x}^{[s]} - \xi_s \cdot \nabla_{\mathbf{x}} L(\mathbf{x}^{[s]})$; for $s = 0, 1, 2, \dots$, until a prespecified termination criterion is met. The parameter $\xi_s > 0$ is the corresponding step-size, which may be adaptive or fixed. To guarantee convergence of L_{ls}, one can select $\xi_s = 2/\sigma_{\max}(\mathbf{A})^2$ for all iterations, though this is too conservative.

C. *Leverage Scores*

Many sampling algorithms select data points according to their *leverage scores* [**?**], [**?**]. 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 defined as $\ell_i := \|\mathbf{U}_{(i)}\|_2^2$, and are agnostic to any particular basis, as they are equal to the diagonal entries of the projection matrix $P_{\mathbf{A}} = \mathbf{AA}^\dagger = \mathbf{UU}^\top$. The *normalized leverage scores* of $\mathbf{A}$ are

$$\pi_i := \left\|\mathbf{U}_{(i)}\right\|_2^2 / \|\mathbf{U}\|_F^2 = \left\|\mathbf{U}_{(i)}\right\|_2^2 / d \qquad \text{for each } i \in \mathbb{N}_N, \tag{4}$$

and $\pi_{\{N\}}$ form a sampling probability distribution; as $\sum_{i=1}^{N} \pi_i = 1$ and $\pi_i \geqslant 0$ for all i. This induced distribution has proven to be useful in linear regression [**?**], [**?**], [**?**], [**?**].

The normalized *block leverage scores*, introduced independently in [**?**], [**?**], are the sum of the normalized leverage scores of the subset of rows constituting the block. Analogous to (**??**), considering the partitioning of $\mathcal{D}$ according to $\mathcal{K}_{\{K\}}$, the *normalized block leverage scores* of $\mathbf{A}$ are defined as

$$\Pi_l := \left\|\mathbf{U}_{(\mathcal{K}_l)}\right\|_F^2 / \|\mathbf{U}\|_F^2 = \left\|\mathbf{U}_{(\mathcal{K}_l)}\right\|_F^2 / d = \sum_{j \in \mathcal{K}_l} \pi_j \quad \text{for each } l \in \mathbb{N}_K. \tag{5}$$

A related notion is that of the *Frobenius block scores*, which in the case of a partitioning as in (**??**); are $\|\mathbf{A}_\iota\|_F^2$ for each $\iota \in \mathbb{N}_K$, which scores have been used for CR-multiplication [**?**], [**?**]. In our context, the block leverage scores of $\mathbf{A}$; are the Frobenius block scores of $\mathbf{U}$.

A drawback of using leverage scores, is that calculating them requires $O(Nd^2)$ time. To alleviate this, one can instead settle for relative-error approximations which can be approximated much faster, *e.g.* [**?**] does so in $O(Nd \log N)$ time. In particular, we can consider approximate normalized scores $\tilde{\Pi}_{\{K\}}$ where $\tilde{\Pi}_i \geqslant \beta \Pi_i$ for all i, for some misestimation factor $\beta \in (0, 1]$. Since $\Pi_{\{K\}}$ and $\tilde{\Pi}_{\{K\}}$ are identical if and only if $\beta = 1$, a higher β implies the approximate distribution is more accurate. When we want to specify that a misestimation factor is for a specific distribution, we accompany it by a corresponding subscript; *e.g.* $\beta_{\tilde{\Pi}}$.

such as the system's limitations, number of servers, or an upper bound on the desired waiting time for the servers to respond. At time T, according to $\tilde{F}(t)$, the central server receives roughly $q(T) := \lfloor \tilde{F}(T) \cdot m \rfloor$ server computations. We refer to the prespecified time instance T after which the central server stops receiving computations; as the "*ending time*". If the sketching procedure of the proposed sketching algorithm were to be carried out by a single server, there would be no benefit in setting T such that $q(T)\tau > N$, as the exact calculation could have taken place in the same amount of time. In distributed networks though there is no control over which servers respond, and it is not a major concern if $q(T)\tau$ is slightly over N; as we still accelerate the computation. The trade-off between accuracy and waiting time t is captured in Theorem **??**, for $q \leftarrow q(t)$ sampling trials. The second hyperparameter we need in order to design an expansion network, is the block size τ; which is determined by K the number of partitions (**??**). Together, $q(T)$ and τ determine the ideal number of servers needed for our framework to perfectly emulate sampling according to the datas' block leverage scores $\Pi_{\{K\}}$.

III. Coded Computing from RandNLA

In this section, we first present our *block* leverage score sampling algorithm, which is more practical and can be carried out more efficiently than its vector-wise counterpart. Our ℓ_2-s.e. result is presented in Theorem **??**. By setting $\tau = 1$ and for $\beta = 1$, we get a known result for (exact) leverage score sampling.

In Subsection **??** we incorporate our block sampling algorithm into the CC probabilistic model described above, in which we leverage task redundancy to mitigate stragglers. Specifically, we show how to replicate computational tasks among the servers, under the integer constraints imposed by the physical system and the desired waiting time; to approximate the gradient at each iteration, in a way that emulates the sampling procedure of the sketch presented in Algorithm **??**. In Subsection **??** we further elaborate on when a perfect emulation is possible, and how emulated block leverage score sampling can be improved when it cannot be done perfectly; through the proposed networks. In Subsection **??** we present our GC approach, and relate it to SD and SSD; which in turn implies convergence guarantees with appropriate step-sizes. Furthermore, at each iteration we have a different induced sketch, hence our procedure lies under the framework of iterative sketching. Specifically, we obtain gradients of multiple sketches of the data $\left(\widetilde{\mathbf{S}}_{[1]}\mathbf{A}, \widetilde{\mathbf{S}}_{[2]}\mathbf{A}, \ldots, \widetilde{\mathbf{S}}_{[n]}\mathbf{A}\right)$ and iteratively refine the solution, where n can be chosen logarithmic in N. A schematic of our approach is provided in Figure **??**, and in Appendix **??** we provide a concrete example of the induced sketching matrices resulting from the iterative process.

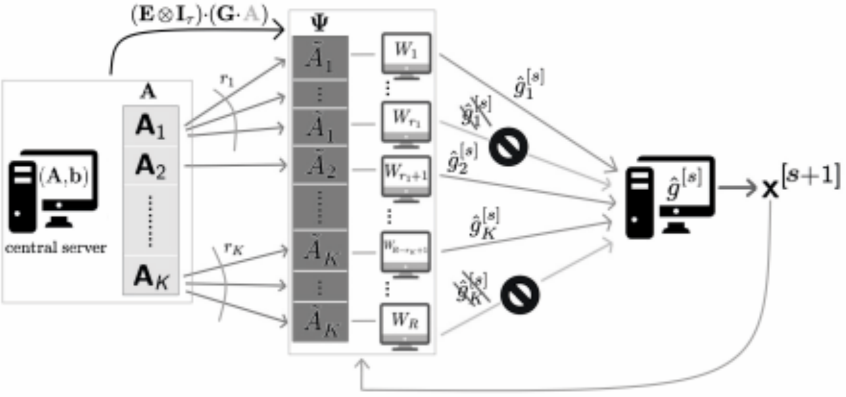

Fig. 2: Illustration of our GC approach, at iteration $s+1$. The blocks of $\mathbf{A}$ (and $\mathbf{b}$) are encoded through $\mathbf{G}$ and then replicated through $\mathbf{E} \otimes \mathbf{I}_\tau$, where each block of the resulting $\mathbf{\Psi}$ is given to a single server. At this iteration, servers W_{r_1} and W_R are stragglers, and their computations are not received. The central server determines the estimate $\hat{g}^{[s]}$, and then shares $\mathbf{x}^{[s+1]}$ with all the servers. The resulting estimate is the gradient of the induced sketch, *i.e.* $\hat{g}^{[s]} = \nabla_{\mathbf{x}} L_{\mathbf{S}}(\widetilde{\mathbf{S}}_{[s]}, \mathbf{A}, \mathbf{b}; \mathbf{x}^{[s]})$.

A. Related Work

Related works [?], [?], [?], [?] have utilized similar ideas to the GC approach we present. The paper titled "Anytime Coding for Distributed Computation" [?] proposes replicating subtasks according to the job, while [?] and [?] incorporate sketching into CC. It is worth noting that even though we focus on gradient methods in this paper; our approach also applies to second-order methods, as well as approximate matrix products through the CR-multiplication algorithm [?], [?], [?]. We briefly discuss this in Section **??**.

The work of [?] deals with matrix-vector multiplication. Similar to our work, they also replicate the computational tasks a certain number of times; and stop the process at a prespecified instance. Here, the computation $\mathbf{A}\mathbf{x}$ for $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $\mathbf{x} \in \mathbb{R}^N$ is broken up into C different tasks; prioritizing the smaller computations. The m servers are split up into c groups, which are asked to compute one of the tasks $\mathbf{y}_j = \left(\sum_{i \in \mathcal{J}_j} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top\right)\mathbf{x}$, for $\mathbf{A} = \sum_{l=1}^N \sigma_l \mathbf{u}_l \mathbf{v}_l^\top$ the SVD representation of $\mathbf{A}$. Each task $\mathbf{y}_j$

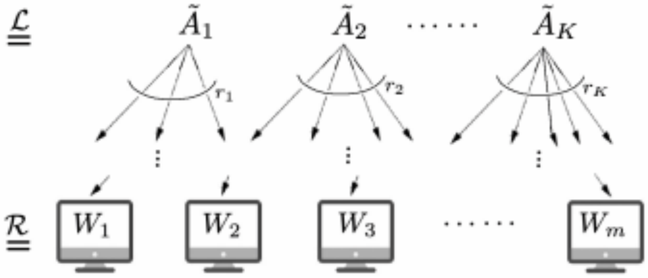
$\underline{\underline{\mathcal{L}}}$
$\tilde{A}_1$
$\tilde{A}_2$
$\tilde{A}_K$
r_1
r_2
r_K
$\underline{\underline{\mathcal{R}}}$
W_1
W_2
W_3
W_m

Fig. 3: Depiction of an expansion network, as a bipartite graph, for $m = \sum_{l=1}^{K} r_l$.

Our goal is to determine $r_{\{K\}}$, which minimize the error in the emulated distribution $\bar{\Pi}_{\{K\}}$. Under the assumption that we have an integer number of replicas per block, from (??) and (??) we deduce that $\Pi_i \approx 1 - \phi(t)^{r_i}$ for $r_i \in \mathbb{Z}_+$, which lead us to the minimization problem[6]

$$\arg \min_{r_{\{K\}} \subsetneq \mathbb{Z}_+} \left\{ \Delta_{\Pi,\bar{\Pi}} \coloneqq \frac{1}{K} \sum_{i=1}^{K} \left|\Pi_i - \left(1 - \phi(t)^{r_i}\right)\right| \right\} = \arg \left\{ \sum_{i=1}^{K} \min_{r_i \in \mathbb{Z}_+} \left\{ \left|\Pi_i - \left(1 - \phi(t)^{r_i}\right)\right| \right\} \right\} . \tag{18}$$

By combining (??) and (??), we then solve for the approximate replications $\hat{r}_{\{K\}}$ at time t:

$$\bar{\Pi}_i \approx \Pi_i = 1 - \phi(t)^{\rho_i(t)} \quad \Longrightarrow \quad \hat{r}_i = \left\lfloor \frac{\log(1 - \Pi_i)}{\log(\phi(t))} \right\rceil = \lfloor \rho_i(t) \rceil , \tag{19}$$

which result in the induced distribution $\bar{\Pi}_i = \hat{r}_i / \hat{R}$, for $\hat{R} := \sum_{l=1}^{K} \hat{r}_l$. In our context, we also require that $\hat{R} \approx m$.

Ideally, the above procedure would result in replication numbers $\hat{r}_{\{K\}}$ for which $\hat{R} = m$. This though is unlikely to occur, as $\Pi_{\{K\}}$ and R are determined by the data, and m is a physical limitation. There are several practical ways to work around this. One approach is to redefine $\hat{r}_{\{K\}}$ to $\tilde{r}_{\{K\}}$ by $\tilde{r}_i = \hat{r}_i \pm \alpha_i$ for α_i small integers such that $\sum_{l=1}^{K} \tilde{r}_l = m$ and $\sum_{l=1}^{K} |\Pi_l - \tilde{r}_l/m|$ is minimal. If $m \gg \hat{R}$ for a large enough τ, we can set the number of replicas to be $\tilde{r}_i \approx \left\lfloor m/\hat{R} \right\rceil \cdot \hat{r}_i$. Furthermore, the block size τ can be selected such that $\hat{R}$ is approximately equal to the system's parameter m. We focus on the issue of having $\hat{R} \approx m$ in Subsection ??.

Lemma 1. *The approximation $\hat{r}_{\{K\}}$ according to* (??) *of the minimization problem* (??) *at time t, satisfies*

$$\Delta_{\Pi,\bar{\Pi}} \leqslant \left(1 - \sqrt{\phi(t)}\right) \cdot \left(\sum_{l=1}^{K} \phi(t)^{\min\limits_{i \in \mathbb{N}_K} \{\hat{r}_i, \rho_i(t)\}} \right) .$$

Proof. We break the proof into the cases where we round $\rho_i(t)$ to both the closest integers above and below. In either case, we know that $\left(\rho_i(t) - \hat{r}_i(t)\right) \in [-1/2, 1/2]$, for each $i \in \mathbb{N}_K$. Denote the respective individual summands of $\Delta_{\Pi,\bar{\Pi}}$ by Δ_i. In the case where $r_i = \lfloor \rho_i(t) \rfloor$, we have $\rho_i(t) = \hat{r}_i + \eta$ for $\eta \in [0, 1/2]$, hence

$$\begin{aligned} \Delta_i &= \left| \left(1 - \phi(t)^{\rho_i(t)}\right) - \left(1 - \phi(t)^{\hat{r}_i}\right) \right| \\ &= \left| \phi(t)^{\hat{r}_i} - \phi(t)^{\rho_i(t)} \right| \\ &= \left| \phi(t)^{\hat{r}_i} - \phi(t)^{r_i + \eta} \right| \\ &= \left| \phi(t)^{\hat{r}_i} \cdot \left(1 - \phi(t)^{\eta}\right) \right| \\ &\leqslant \left| \phi(t)^{\hat{r}_i} \cdot \left(1 - \phi(t)^{1/2}\right) \right| \\ &= \phi(t)^{\hat{r}_i} \cdot \left(1 - \sqrt{\phi(t)}\right) . \end{aligned}$$

Similarly, in the case where $r_i = \lceil \rho_i(t) \rceil$, we have $\hat{r}_i = \rho_i(t) + \eta$ for $\eta \in [0, 1/2]$; and

$$\Delta_i \leqslant \phi(t)^{\rho_i(t)} \cdot \left(1 - \sqrt{\phi(t)}\right) .$$

Considering all summands, it follows that

$$\Delta_{\Pi,\bar{\Pi}} = \sum_{l=1}^{K} \Delta_l \leqslant \sum_{l=1}^{K} \left(\phi(t)^{\min\limits_{i \in \mathbb{N}_K} \{\hat{r}_i, \rho_i(t)\}} \cdot \left(1 - \sqrt{\phi(t)}\right) \right) .$$

[6] Note that $\Delta_{\Pi,\bar{\Pi}} \equiv d_{\Pi,\tilde{\Pi}}$, where $\tilde{\Pi}_i = 1 - \phi(t)^{r_i}$ for all $i \in \mathbb{N}_K$. For our proposed distribution $\bar{\Pi}_{\{K\}}$, we may have $d_{\Pi,\bar{\Pi}} \neq \Delta_{\Pi,\bar{\Pi}}$.

modified problem (**??**) which only accounts for a reduced dimension determined at the beginning of the iterative process, whose approximate solution is $\mathbf{x}^\star_{\mathbf{G}} \approx (\widetilde{\mathbf{S}}\mathbf{A})^\dagger(\widetilde{\mathbf{S}}\mathbf{b})$.[8] Instead, we consider a different reduced linear system $\widetilde{\mathbf{S}}_{[s]} \cdot \left(\mathbf{A}\mathbf{x}^{[s]}\right) = \widetilde{\mathbf{S}}_{[s]} \cdot \mathbf{b}$ at each iteration. This further justifies the result of Corollary **??**. This benefit of iterative sketching is validated numerically in Section **??**.

Theorem 2. *The proposed GC scheme based on the block leverage score sketch results in a SSD procedure for $L_{ls}(\mathbf{\Psi}, \vec{\psi}; \mathbf{x})$. Furthermore, at each iteration it produces an unbiased estimator of* (**??**)*, i.e.* $\mathbb{E}\left[\hat{g}^{[s]}\right] = g^{[s]}$.

Proof. The computations of the q fastest servers indexed by $\mathcal{I}^{[s]}$ (which corresponds to $\widetilde{\mathbf{\Omega}}^{[s]}$), are added to produce $\hat{g}^{[s]}$, and the sampling of Algorithm **??** is according to $\bar{\Pi}_{\{K\}}$. By Remark **??**, it follows that each $\mathcal{I}^{[s]}$ has equal chance of occurring, which is precisely the stochastic step of SSD, *i.e.* each group of q encoded block pairs has an equal chance of being selected.

Since the servers are homogeneous and respond independently of each other, it follows that at iteration s; each $\hat{g}_i$ is received with probability $\bar{\Pi}_i$. Therefore

$$\begin{aligned}\mathbb{E}\left[\hat{g}^{[s]}\right] &= \mathbb{E}\left[\sum_{i\in\mathcal{I}^{[s]}} \hat{g}_i^{[s]}\right] = \sum_{i\in\mathcal{I}^{[s]}} \mathbb{E}\left[\hat{g}_i^{[s]}\right] = \sum_{i\in\mathcal{I}^{[s]}} \sum_{j=1}^{K} \bar{\Pi}_j \cdot \hat{g}_j^{[s]} \\ &= q \cdot \sum_{j=1}^{K} \bar{\Pi}_j \cdot \hat{g}_j^{[s]} \overset{\flat}{=} q \cdot \sum_{j=1}^{K} \bar{\Pi}_j \cdot \frac{1}{q\bar{\Pi}_j} \cdot g_j^{[s]} = \sum_{j=1}^{K} g_j^{[s]} = g^{[s]}\end{aligned}$$

where in $\flat$ we invoked (**??**). □

Lemma 3. *The optimal solution of the modified least squares problem $L_{ls}(\mathbf{\Psi}, \vec{\psi}; \mathbf{x})$, is equal to the optimal solution $\mathbf{x}^\star$ of* (**??**)*.*

Proof. Note that the modified objective function $L_{ls}(\mathbf{\Psi}, \vec{\psi}; \mathbf{x})$ is $\|\widetilde{\mathbf{E}}\mathbf{G} \cdot (\mathbf{A}\mathbf{x} - \mathbf{b})\|_2^2$. Denote its optimal solution by $x^\star \in \mathbb{R}^d$. Further note that $\widetilde{\mathbf{E}}$ is comprised of $\tau \times \tau$ identity matrices in such a way that it is full-rank, and $\mathbf{G}$ corresponds to a rescaling of these $\mathbf{I}_\tau$ matrices, thus $\widetilde{\mathbf{E}}\mathbf{G}$ is also full-rank. It then follows that

$$x^\star = \left((\mathbf{EG}) \cdot \mathbf{A}\right)^\dagger \cdot \left((\mathbf{EG}) \cdot \mathbf{b}\right) = \mathbf{A}^\dagger \cdot \left((\mathbf{EG})^\dagger \cdot (\mathbf{EG})\right) \cdot \mathbf{b} = \mathbf{A}^\dagger \cdot \mathbf{I}_N \cdot \mathbf{b} = \mathbf{x}^\star.$$

□

The crucial aspect of our expansion network (incorporated in Theorem **??**), which allowed us to use block leverage score sampling in the proposed GC scheme, is that uniform sampling of $L_{ls}(\mathbf{\Psi}, \vec{\psi}; \mathbf{x}^{[s]})$ is $\beta_{\bar{\Pi}}$-approximately equivalent to block sampling of $L_{ls}(\tilde{\mathbf{A}}, \tilde{\mathbf{b}}; \mathbf{x}^{[s]})$ according to the block leverage scores of $\mathbf{A}$. Since the two objective functions are differentiable and additively separable, the resulting gradients are equal, under the assumption that we use the same $\mathbf{x}^{[s]}$ and sampled index set $\mathcal{S}^{[s]}$.[9] As previously mentioned, the main drawback is that in certain cases we need significantly more servers to accurately emulate $\Pi_{\{K\}}$.

The significance of Theorem **??**, is that our distributed approach guarantees well-known established SD and SSD results which assume that the approximate gradient is an unbiased estimator, *e.g.* [**?**, Chapter 14]. Even though we are not guaranteed a descent at every iteration (*i.e.* we could have $L_{ls}(\mathcal{D}; \mathbf{x}^{[s+1]}) > L_{ls}(\mathcal{D}; \mathbf{x}^{[s]})$ or $\|\mathbf{x}^{[s+1]} - \mathbf{x}^\star\|_2^2 > \|\mathbf{x}^{[s]} - \mathbf{x}^\star\|_2^2$), stochastic descent methods are more common in practice when dealing with large datasets, as empirically they outperform regular SD. This is also confirmed in our experiments.

F. Convergence to $\mathbf{x}^\star$

Next, we give a summary of our main results thus far, and explain how together they imply convergence of our approach in expectation, to iteratively solves $L_{ls}(\mathbf{\Psi}, \vec{\psi}; \mathbf{x}^{[s]})$. Moreover, the contraction of our method is quantified in Appendix **??**.

Firstly, as summarized in Remark **??**, we mimic block leverage score sampling w.r. of $(\mathbf{A}, \mathbf{b})$ (from $L_{ls}(\mathbf{A}, \mathbf{b}; \mathbf{x}^{[s]})$) through uniform sampling, by approximately solving (**??**) through the implication of (**??**) (Lemma **??**). This is done implicitly by communicating computations over erasure channels. Secondly, by Theorem **??** we know that the proposed block leverage score sketching matrices satisfy (**??**); where the approximate sampling distribution $\bar{\Pi}_{\{K\}}$ is determined through the proposed expansion network associated with $\Pi_{\{K\}}$. Hence, at each iteration, we approach a solution $\hat{\mathbf{x}}^{[s]}$ of the induced sketched system $\widetilde{\mathbf{S}}_{[s]} \cdot \left(\mathbf{A}\mathbf{x}^{[s]}\right) = \widetilde{\mathbf{S}}_{[s]} \cdot \mathbf{b}$, which $\widetilde{\mathbf{S}}_{[s]}$ satisfies (**??**) with overwhelming probability. Thirdly, by Theorem **??** and Lemma **??**, with a diminishing step-size ξ_s, our updates $\mathbf{x}^{[s]}$ converge to $\mathbf{x}^\star$ in expectation, at a rate of $O(1/\sqrt{s} + r/s)$ [**?**], [**?**]. A synopsis is given below:

$$\left\{\begin{array}{c} L_{\mathbf{S}}(\widetilde{\mathbf{S}}_{[s]}, \mathbf{A}, \mathbf{b}; \mathbf{x}) \text{ sol'ns} \\ \text{satisfy } (\mathbf{??}) \text{ and } (\mathbf{??}) \end{array}\right\} \xleftarrow{??,??} \left\{\begin{array}{c} \text{Solve } L_{ls}(\mathbf{\Psi}, \vec{\psi}; \mathbf{x}^{[s]}) \\ \text{through `sketched-GC'} \end{array}\right\} \xrightarrow{??,??} \left\{\begin{array}{c} \text{With a diminishing } \xi_s\text{:} \\ \lim \mathbb{E}[\mathbf{x}^{[s]}] \to \mathbf{x}^\star \end{array}\right\}.$$

[8] Since $\widetilde{\mathbf{S}} \in \mathbb{R}^{r \times N}$ for $r < N$; we have $\widetilde{\mathbf{S}}^\dagger \widetilde{\mathbf{S}} \neq \mathbf{I}_N$, hence $(\widetilde{\mathbf{S}}\mathbf{A})^\dagger(\widetilde{\mathbf{S}}\mathbf{b}) \neq \mathbf{A}^\dagger\mathbf{b}$.

[9] The index set of the sampled blocks from $L_{ls}(\mathbf{\Psi}, \vec{\psi}; \mathbf{x}^{[s]})$, corresponds to an index *multiset* of the sampled blocks from $L_{ls}(\tilde{\mathbf{A}}, \tilde{\mathbf{b}}; \mathbf{x}^{[s]})$, as in the latter we are considering sampling *with replacement*.

where in $\flat$ we make use of the assumption that $\mathbf{S}$ satisfies (**??**). Our approximate GC approach therefore (w.h.p.) satisfies (**??**), with $\mathrm{err}(\mathbf{G}_{(\mathcal{I})}) = \epsilon/\sqrt{K}$. □

IV. Experiments

In this section, we corroborate our theoretical results, and show their benefits on fabricated datasets. The minimum benefit of Algorithm **??** occurs when $\Pi_{\{K\}}$ is close to uniform. For this reason, and the fact that our expansion approach depends on the implicit distribution through $r_{\{K\}}$, we construct dataset matrices whose resulting sampling distributions and block leverage scores are non-uniform.

For the first experiment, we considered $\mathbf{A} \in \mathbb{R}^{2000\times 40}$ following a t-distribution, and standard Gaussian noise was added to an arbitrary vector from $\mathrm{im}(\mathbf{A})$ to define $\mathbf{b}$. We considered $K = 100$ blocks, thus $\tau = 20$. The effective dimension $N = 2000$ was reduced to $r = 1000$, *i.e.* $q = 20$. We compared the iterative approach with exact block leverage scores (*i.e.* $\beta = 1$), against analogous approaches using the block-SRHT and Gaussian sketches, and uncoded regular SD.

In Figure **??** we ran 600 iterations on six different instances for each approach, and varied ξ for each experiment by logarithmic factors of $\xi^{\times} = 2/\sigma_{\max}(\mathbf{A})^2$. The average log residual errors $\log_{10}\left(\|\mathbf{x}^\star_{l_s} - \hat{\mathbf{x}}\|_2/\sqrt{N}\right)$ are depicted in Figure **??**, and reported in Table **??**. In Figure **??** we observe the convergence of the different approaches, in the case where $\xi \approx 0.42$. In this case, our method (block-lvg) outperforms the Gaussian sketching approach and regular SD. The fact that the performance of the block-SRHT is similar to our proposed algorithm, reflects the result of Proposition **??**.

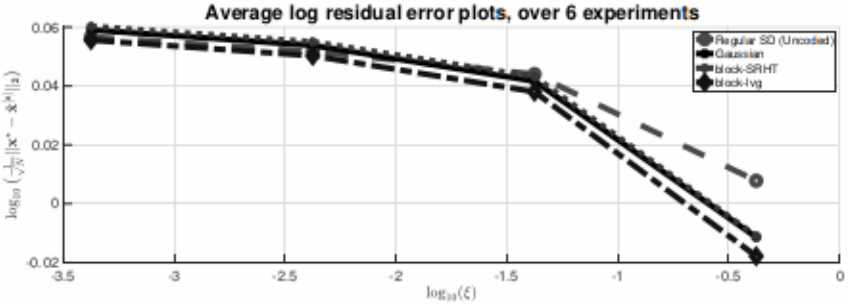

Fig. 4: residual error for varying ξ_s

Average log **residual error** $\log_{10}\left(\|\mathbf{x}^\star_{l_s} - \hat{\mathbf{x}}\|_2/\sqrt{N}\right)$				
$\log_{10}(\xi)$	0.0004	0.0042	0.0421	0.4207
Regular SD	0.0566	0.0517	0.0440	0.0078
Gaussian	0.0590	0.0538	0.0416	-0.0114
block-SRHT	0.0603	0.0550	0.0431	-0.0110
block-lvg	0.0556	0.0502	0.0380	-0.0178

TABLE I: Average log residual errors, for six instances of SD with fixed steps, when performing Gaussian sketching with updated sketches, iterative block-SRHT and iterative block leverage score sketching, and uncoded SD.

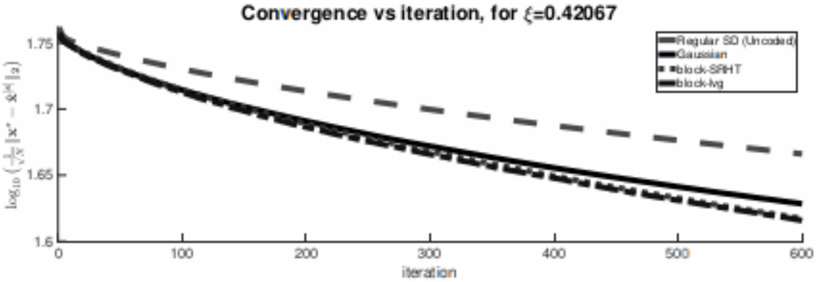

Fig. 5: log residual error convergence

We also considered the same experiment with $\mathbf{A}$ drawn from a t-distribution, with and optimal step-size $\xi^\star_s = \langle \mathbf{A}g^{[s]}, \mathbf{A}\mathbf{x}^{[s]} - \mathbf{b}\rangle/\|\mathbf{A}g^{[s]}\|_2^2$ at each iteration. From Figure **??**, we observe that our iterative sketching approach outperforms Gaussian sketching with updated sketches; and iterative sketching is superior to non-iterative. Furthermore, we validate Lemma **??** and Theorem **??**, as our iterative sketching approach and SSD have similar convergence. Furthermore, it was observed that in some case cases when our iterative sketching method would outperform regular SD (and SSD). We also compared our method to iterative and non-iterative approaches according to the block leverage score sampling, block-SRHT, and Rademacher sketching methods, in which our corresponding approach again produced more accurate final approximations.

$$
\begin{aligned}
&= \left\| (d/\beta - 1)\cdot \mathbf{I}_d - \frac{d}{\beta}\left(\sum_{\iota=1}^{K} \mathbf{U}_{(\mathcal{K}_\iota)}^\top \overbrace{\left(\mathbf{U}_{(\mathcal{K}_\iota)}\mathbf{U}_{(\bar{\mathcal{K}}_\iota)}^\top\right)}^{\mathbf{0}_{d\times d}} \mathbf{U}_{(\bar{\mathcal{K}}_\iota)} \right) \right\|_2 \\
&= \|(d/\beta - 1)\cdot \mathbf{I}_d\|_2 \\
&= d/\beta - 1
\end{aligned}
$$

where in $\sharp$ we used the fact $\Pi_\iota/\tilde{\Pi}_\iota \leqslant 1/\beta$, in $\flat$ that $\|\mathbf{U}_{(\mathcal{K}_\iota)}\|_2^2 = 1$, and in $\natural$ that $\mathbf{U}_{(\mathcal{K}_\iota)}^\top \mathbf{U}_{(\mathcal{K}_\iota)} = \mathbf{I}_d - \mathbf{U}_{(\bar{\mathcal{K}}_\iota)}^\top \mathbf{U}_{(\bar{\mathcal{K}}_\iota)}$ for each ι.

According to Theorem **??**, we substitute $\gamma = 1 + d$ and $\sigma^2 = d/\beta - 1$ to get

$$
\begin{aligned}
\frac{1}{q}\sum_{i=1}^{q}\mathbf{X}_i &= \frac{1}{q}\sum_{i=1}^{q}\left(\mathbf{I}_d - \frac{\mathbf{U}_{(\mathcal{K}_{j(i)})}^\top \mathbf{U}_{(\mathcal{K}_{j(i)})}}{\tilde{\Pi}_{j(i)}}\right) \\
&= \mathbf{I}_d - \frac{1}{q}\left(\sum_{i=1}^{q}\frac{\mathbf{U}_{(\mathcal{K}_{j(i)})}^\top \mathbf{U}_{(\mathcal{K}_{j(i)})}}{\tilde{\Pi}_{j(i)}}\right) \\
&= \mathbf{I}_d - \mathbf{U}^\top \widetilde{\mathbf{S}}^\top \widetilde{\mathbf{S}}\mathbf{U}
\end{aligned}
$$

$$= \mathbf{I}_d - \mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\mathbf{U}$$

where the last equality follows from the definition of $\widetilde{\mathbf{S}}$. Putting everything together into Theorem **??**, we get that

where the last equality follows from the definition of $\mathbf{S}$. Putting everything together into Theorem **??**, we get that

$$
\Pr\left[\|\mathbf{I}_d - \mathbf{U}^\top \widetilde{\mathbf{S}}^\top \widetilde{\mathbf{S}}\mathbf{U}\|_2 > \epsilon\right] \leqslant 2d \cdot e^{-q\epsilon^2\Theta(\beta/d)}.
$$

$$
\Pr\left[\|\mathbf{I}_d - \mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\mathbf{U}\|_2 > \epsilon\right] \leqslant 2d \cdot e^{-q\epsilon^2\Theta(\beta/d)}.
$$

□

Proof. [Theorem **??**] By substituting $q = \Theta\left(\frac{d}{\tau}\log\left(2d/\delta\right)/\left(\beta\epsilon^2\right)\right)$ in (**??**) and taking the complementary event, we attain the desired statement. □

Before we prove Corollary **??**, we introduce the notion of *block α-balanced*, which is a generalization of *α-balanced* from [**?**]. The sampling distribution $\Pi_{\{K\}}$ is said to be *block α-balanced*, if

$$
\max_{i\in\mathbb{N}_K}\{\Pi_i\} \leqslant \frac{\alpha}{N/\tau} = \frac{\alpha}{K} \tag{31}
$$

for some constant α independent of K and q. Furthermore, in our context, if the individual leverage scores $\pi_{\{N\}}$ are α-balanced for α independent of N and τ, then the block leverage scores $\Pi_{\{K\}}$ are block α-balanced, as

$$
\max_{i\in\mathbb{N}_K}\{\Pi_i\} \leqslant \tau \cdot \max_{j\in\mathbb{N}_N}\{\pi_j\} \leqslant \tau \cdot \frac{\alpha}{N} = \frac{\alpha}{N/\tau} = \frac{\alpha}{K}\,. \tag{32}
$$

Proof. [Corollary **??**] From the proof of [**?**, Theorem 1], we simply need to show that

$$
\left\|\mathbb{E}\left[\widetilde{\mathbf{S}}^\top\left(\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^\top\right)^{-1}\widetilde{\mathbf{S}}\right]\right\|_2 \leqslant \eta \cdot \frac{\tau}{N}
$$

for $\widetilde{\mathbf{S}}$ a single sketch produced in Algorithm **??**, and an appropriate constant η independent of N and τ. We assume that the individual leverage scores $\pi_{\{N\}}$ are α-balanced, where α is a constant independent of N and τ. By (**??**), it follows that the block leverage scores $\Pi_{\{K\}}$ are block α-balanced; *i.e.* $\Pi_i \leqslant \Pi_K \leqslant \frac{\alpha}{K}$ for all $i \in \mathbb{N}_{K=1}$.

A direct computation shows that

$$
\begin{aligned}
\left(\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^\top\right)^{-1} &= \Big((\mathbf{D}\cdot\mathbf{\Omega}\otimes\mathbf{I}_\tau)\cdot(\mathbf{\Omega}^\top\cdot\mathbf{D}^\top\otimes\mathbf{I}_\tau)\Big)^{-1} \\
&= \Big((\mathbf{D}\cdot\mathbf{\Omega}\cdot\mathbf{\Omega}^\top\cdot\mathbf{D}^\top)\otimes\mathbf{I}_\tau\Big)^{-1} \\
&= (\mathbf{D}\cdot\mathbf{\Omega}\cdot\mathbf{\Omega}^\top\cdot\mathbf{D}^\top)^{-1}\otimes\mathbf{I}_\tau
\end{aligned}
$$

and consequently

$$
\begin{aligned}
\widetilde{\mathbf{S}}^\top\left(\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^\top\right)^{-1}\widetilde{\mathbf{S}} &= (\mathbf{\Omega}^\top\cdot\mathbf{D}^\top\otimes\mathbf{I}_\tau)\cdot\Big((\mathbf{D}\cdot\mathbf{\Omega}\cdot\mathbf{\Omega}^\top\cdot\mathbf{D}^\top)^{-1}\otimes\mathbf{I}_\tau\Big)\cdot(\mathbf{D}\cdot\mathbf{\Omega}\otimes\mathbf{I}_\tau) \\
&= \Big(\mathbf{\Omega}^\top\cdot\mathbf{D}^\top\cdot(\mathbf{D}\cdot\mathbf{\Omega}\cdot\mathbf{\Omega}^\top\cdot\mathbf{D}^\top)^{-1}\otimes\mathbf{I}_\tau\Big)\cdot(\mathbf{D}\cdot\mathbf{\Omega}\otimes\mathbf{I}_\tau) \\
&= \underbrace{\Big(\mathbf{\Omega}^\top\cdot\mathbf{D}^\top\cdot(\mathbf{D}\cdot\mathbf{\Omega}\cdot\mathbf{\Omega}^\top\cdot\mathbf{D}^\top)^{-1}\cdot\mathbf{D}\cdot\mathbf{\Omega}\Big)}_{:=Z\in\mathbb{R}_{\geqslant 0}^{K\times K}}\otimes\mathbf{I}_\tau
\end{aligned}
$$

We note that since $\mathbf{\Psi}$ has repeated blocks from the expansion, the scores we consider in Lemma **??** are not the block leverage scores of $\mathbf{\Psi}$. The Frobenius block scores of $\tilde{\mathbf{U}}_{\text{exp}}$, are in fact the corresponding block leverage scores of $\tilde{\mathbf{A}}$, which are replicated in the expansion. Moreover, note that the closer $\beta_{\tilde{\Pi}}$ is to 1, the closer the sampling distribution according to the Frobenius block scores of $\tilde{\mathbf{U}}_{\text{exp}}$; which we denote by $\tilde{Q}_{\{R\}}$, is to being exactly uniform. We denote the uniform sampling distribution by $\mathcal{U}_{\{R\}}$.

Lemma 4. *When $\tilde{\Pi}_{\{K\}} = \Pi_{\{K\}}$, the sampling distribution $\tilde{Q}_{\{R\}}$ is uniform. When $\tilde{\Pi}_\iota \geqslant \beta_{\tilde{\Pi}}\Pi_\iota$ for $\beta_{\tilde{\Pi}} = \min_{i\in\mathbb{N}_K}\{\Pi_i/\tilde{\Pi}_i\} \in (0,1)$ and all $\iota \in \mathbb{N}_K$, the resulting distribution $\tilde{Q}_{\{R\}}$ is approximately uniform, and satisfies $d_{\mathcal{U},\tilde{Q}} \leqslant 1/(R\beta_{\tilde{\Pi}})$.*

Proof. Let $\mathbf{U} = \begin{bmatrix}\mathbf{U}_1^\top & \cdots & \mathbf{U}_K^\top\end{bmatrix}^\top$ denote the corresponding blocks of $\mathbf{U}$ according the partitioning of $\mathcal{D}$. Without loss of generality, assume that the data points within each partition are consecutive rows of $\mathbf{A}$, and $\mathbf{U}_\iota \in \mathbb{R}^{\tau\times d}$ for all $\iota \in \mathbb{N}_K$.

From (**??**) and (**??**), it follows that

$$\begin{aligned}
\mathbf{\Psi} = \widetilde{\mathbf{E}}\cdot\tilde{\mathbf{A}} &= (\mathbf{E}\otimes\mathbf{I}_\tau)\cdot\big(\mathbf{G}\cdot\mathbf{U\Sigma V}^\top\big)\\
&= (\mathbf{E}\otimes\mathbf{I}_\tau)\cdot\Big[\mathbf{U}_1^\top/\sqrt{q\tilde{\Pi}_1}\ \cdots\ \mathbf{U}_K^\top/\sqrt{q\tilde{\Pi}_K}\Big]^\top\cdot\mathbf{\Sigma V}^\top\\
&=: (\mathbf{E}\otimes\mathbf{I}_\tau)\cdot\Big[\tilde{\mathbf{U}}_1^\top\ \cdots\ \tilde{\mathbf{U}}_K^\top\Big]^\top\cdot\mathbf{\Sigma V}^\top\\
&=: \overbrace{\Big[\underbrace{\tilde{\mathbf{U}}_1^\top\ \cdots\ \tilde{\mathbf{U}}_1^\top}_{r_1}\ \underbrace{\tilde{\mathbf{U}}_2^\top\ \cdots\ \tilde{\mathbf{U}}_2^\top}_{r_2}\ \cdots\ \underbrace{\tilde{\mathbf{U}}_K^\top\ \cdots\ \tilde{\mathbf{U}}_K^\top}_{r_K}\Big]^\top}^{\tilde{\mathbf{U}}_{\text{exp}}\in\mathbb{R}^{R\tau\times d}}\cdot\mathbf{\Sigma V}^\top.
\end{aligned}$$

Note that $\tilde{\mathbf{U}}_{\text{exp}}\mathbf{\Sigma V}^\top$ is not the SVD of $\mathbf{\Psi}$. For the normalizing factor of $\frac{q}{Rd}$:

$$\tilde{Q}_\iota = \frac{q}{Rd}\cdot\|\tilde{\mathbf{U}}\|_F^2 = \frac{q}{Rd}\cdot\frac{\|\mathbf{U}_\iota\|_F^2}{q\tilde{\Pi}_\iota} = \frac{\Pi_\iota}{R\tilde{\Pi}_\iota} \leqslant \frac{1}{R\beta_{\tilde{\Pi}}} \quad\Longrightarrow\quad \sum_{i=1}^{R}\big|\tilde{Q}_i - 1/R\big| \overset{\triangle}{\leqslant} \frac{R}{R\beta_{\tilde{\Pi}}} = \frac{1}{\beta_{\tilde{\Pi}}}\ ,$$

where $\triangle$ follows from the fact that $\big|\tilde{Q}_i - 1/R\big| \leqslant 1/\big(R\beta_{\tilde{\Pi}}\big)$ for each $i \in \mathbb{N}_R$. After normalizing by $1/R$ according to the distortion metric, we deduce that $d_{\mathcal{U},\tilde{Q}} \leqslant 1/\big(R\beta_{\tilde{\Pi}}\big)$.

In the case where $\tilde{\Pi}_{\{K\}} = \Pi_{\{K\}}$, we have

$$\tilde{Q}_\iota = \frac{q}{Rd}\cdot\|\tilde{\mathbf{U}}_\iota\|_F^2 = \frac{q}{Rd}\cdot\frac{\|\mathbf{U}_\iota\|_F^2}{q\Pi_\iota} = \frac{\Pi_\iota}{R\Pi_\iota} = \frac{1}{R}$$

for all $\iota \in \mathbb{N}_K$, thus $\tilde{Q}_{\{K\}} = \mathcal{U}_{\{K\}}$. □

Finally, in Proposition **??** we show when the block leverage score sampling sketch of Algorithm **??** and the block-SRHT of [**?**] have the same ℓ_2-s.e. guarantee. We first recall the corresponding result to Theorem **??**, of the block-SRHT.

Theorem 5 ([**?**, Theorem 7]). *The block-SRHT $\mathbf{S}_{\hat{\mathbf{H}}}$ is a ℓ_2-s.e. of $\mathbf{A}$. For $\delta > 0$ and $q = \Theta\big(\frac{d}{\tau}\log(Nd/\delta)\cdot\log(2d/\delta)/\epsilon^2\big)$:*

$$\Pr\Big[\|\mathbf{I}_d - \mathbf{U}^\top\mathbf{S}_{\hat{\mathbf{H}}}^\top\mathbf{S}_{\hat{\mathbf{H}}}\mathbf{U}\|_2 \leqslant \epsilon\Big] \geqslant 1-\delta\ .$$

Proposition 5. *Let $\beta = 1$. For $\delta = e^{\Theta(1)}/(Nd)$, the sketches of Algorithm **??** and the block-SRHT of [?] achieve the same asymptotic ℓ_2-s.e. guarantee, for the same number of sampling trials q.*

Proof. For $\delta = e^{\Theta(1)}/(Nd)$, the two sketching methods have the same q, and both satisfy the ℓ_2-s.e. property with error probability $1-\delta$. □

Appendix D
Contraction Rate of Block Leverage Score Sampling

In this appendix we quantify the contraction rate of our method on the error term $\mathbf{x}^{[s]} - \mathbf{x}^\star$, which further characterizes the convergence of SD after applying our method. The contraction rate is compared to that of regular SD.

Recall that the contraction rate of an iterative process given by a function $h(x^{[s]})$ is the constant $\gamma \in (0,1)$ for which at each iteration we are guaranteed that $h(x^{[s+1]}) \leqslant \gamma\cdot h(x^{[s]})$, therefore $h(x^{[s]}) \leqslant \gamma_s\cdot h(x^{[0]})$. Let ξ be a fixed step-size, $\widetilde{\mathbf{S}}_{[s]}$ the induced sketching matrix of Algorithm **??** at iteration s, and define $\mathbf{B}_{SD} = \big(\mathbf{I}_d - 2\xi\cdot\mathbf{A}^\top\mathbf{A}\big)$ and $\mathbf{B}_s = \Big(\mathbf{I}_d - 2\xi\cdot\mathbf{A}^\top\widetilde{\mathbf{S}}_{[s]}^\top\widetilde{\mathbf{S}}_{[s]}\mathbf{A}\Big)$. We note that the contraction rates could be further improved if one also incorporates an optimal step-size. It is also worth noting that when weighting from Appendix **??** is introduced, we have the same contraction rate and straggler ratio.

Lemma 5. *For $\widetilde{\mathbf{S}}$ the sketching matrix of Algorithm **??**, we have $\mathbb{E}\Big[\widetilde{\mathbf{S}}^\top\widetilde{\mathbf{S}}\Big] = \mathbf{I}_N$.*