Appendices

Contents

A Proof of Preliminaries (Section 3)

A.1 Norms (Lemma 1)

Lemma 1. *For a matrix $A \in \mathbb{R}^{d \times n}$ and $p \in [1,2)$, $\|A\|_{p,2} \leq \|A\|_p \leq d^{\frac{1}{p}-\frac{1}{2}} \|A\|_{p,2}$.*

Proof. Let $x \in \mathbb{R}^d$. For $0 < p < r$,

$$\|x\|_r \leq \|x\|_p \leq d^{\frac{1}{p}-\frac{1}{r}} \|x\|_r$$

Let $r = 2$. Then we have

$$\|x\|_2 \leq \|x\|_p \leq d^{\frac{1}{p}-\frac{1}{2}} \|x\|_2$$

Note that $\|A\|_{p,2} = \left(\sum_j \|A_{*j}\|_2^p\right)^{\frac{1}{p}}$ and $\|A\|_p = \left(\sum_j \|A_{*j}\|_p^p\right)^{\frac{1}{p}}$.

Therefore,

$$\|A\|_{p,2} = \left(\sum_j \|A_{*j}\|_2^p\right)^{\frac{1}{p}} \leq \left(\sum_j \|A_{*j}\|_p^p\right)^{\frac{1}{p}} = \|A\|_p$$

and

$$\|A\|_p = \left(\sum_j \|A_{*j}\|_p^p\right)^{\frac{1}{p}} \leq d^{\frac{1}{p}-\frac{1}{2}} \left(\sum_j \|A_{*j}\|_2^p\right)^{\frac{1}{p}} = d^{\frac{1}{p}-\frac{1}{2}} \|A\|_{p,2}$$

□

A.2 Sketched Error Lower Bound (Lemma 2)

We show a lower bound on the approximation error for a sketched subset of columns, $\|SA_TV - SA\|_p$, in terms of $\|A_TV - A\|_p$. The lower bound holds simultaneously for any arbitrary subset A_T of chosen columns, and for any arbitrary right factor V.

We begin the proof by first showing that applying a dense p-stable sketch to a vector will not shrink its p-norm. This is done in **Lemma** 2.1. We further observe that although p-stable random variables are heavy-tailed, we can still bound their tail probabilities by applying Lemma 9 from [1]. We note this in **Lemma** 2.2. Note that the X_i's do not need to be independent in this lemma.

Equipped with **Lemma** 2.1, **Lemma** 2.2 and a net argument, we can now establish a lower bound on $\|SA_TV - SA\|_p$. We first show in **Lemma** 2.3 that, with high probability, for any arbitrarily selected subset A_T of columns and for an arbitrary column A_{*j}, the error incurred to fit SA_{*j} using the columns of SA_T is no less than the error incurred to fit A_{*j} using the columns of A_T. We then apply a union bound over all subsets $T \subset [n]$ and columns $j \in [n]$ to conclude our lower bound in **Lemma** 2.

Lemma 2.1. *(No Contraction of p-stable Sketch) Given a matrix $S \in \mathbb{R}^{t \times m}$ whose entries are i.i.d. p-stable random variables rescaled by $\Theta\left(\frac{1}{t^{\frac{1}{p}}}\right)$, where $1 \leq p < 2$, for any fixed $y \in \mathbb{R}^m$, with probability $1 - \frac{1}{e^t}$, the following holds:*

$$\|Sy\|_p \geq \|y\|_p$$

Proof. By p-stability, we have $\|Sy\|_p^p = \sum_{i=1}^t \left(\|y\|_p \frac{|Z_i|}{t^{\frac{1}{p}}}\right)^p$, where the Z_i are i.i.d. p-stable random variables. Since $Pr[|Z_i| = \Omega(1)] > \frac{1}{2}$, by applying a Chernoff bound (to the indicators $1_{|Z_i| \geq C}$ for a sufficiently small constant C), we have $\sum_{i=1}^t |Z_i|^p = \Omega(t)$ with probability $1 - \frac{1}{e^t}$. Therefore, with probability $1 - \frac{1}{e^t}$, $\|Sy\|_p \geq \|y\|_p$. □

Lemma 2.2. *(Upper Tail Inequality for p-stable Distributions) Let $p \in (1,2)$, and $m > 3$. For $i \in [m]$, let X_i be a standard p-stable random variable, and let $\gamma_i > 0$ and $\gamma = \sum_{i=1}^m \gamma_i$. Let $X = \sum_{i=1}^m \gamma_i |X_i|^p$. Then, for any $t \geq 1$, $Pr[X \geq t\alpha_p\gamma] \leq \frac{2\log(mt)}{t}$, where $\alpha_p > 0$ is a constant that is at most 2^{p-1}.*

Proof. Lemma 9 from [1] for $p \in (1,2)$. □

Lemma 2.3. *(No Contraction for All Sketched Subsets and Columns) Let $A \in \mathbb{R}^{d \times n}$, and $k \in \mathbb{N}$. Let $t = k \cdot poly(\log nd)$, and let $S \in \mathbb{R}^{t \times d}$ be a matrix whose entries are i.i.d. standard p-stable random variables, rescaled by $\Theta(1/t^{\frac{1}{p}})$. Finally, let $m = k \cdot poly(\log k)$. Then, with probability $1 - \frac{1}{poly(nd)}$, for all $T \subset [n]$ with $|T| = m$, for all $j \in [n]$, and for all $y \in \mathbb{R}^{|T|}$,*

$$\|A_T y - A_{*j}\|_p \leq \|S(A_T y - A_{*j})\|_p$$

Proof. **Step 1:** We first extend **Lemma** 2.1 and use a net argument to show that applying a p-stable sketching matrix $S \in \mathbb{R}^{t \times d}$ will not shrink the norm of *any* vector, i.e. $\|Sy\|_p \geq \|y\|_p$ simultaneously for *all* y in the column span of $[A_T, A_j] =: A_{T,j}$, for any fixed $T \subset [n]$ with $|T| = k \cdot \text{poly}(\log k)$, and $j \in [n]$.

For our net argument, we begin by showing that with high probability all entries of S are bounded. Let $D > 0$, which we will choose appropriately later. For convenience, let $\widetilde{S} \in \mathbb{R}^{t \times n}$ be equal to S without the rescaling by $\Theta(1/t^{1/p})$ (that is, the entries of $\widetilde{S}$ are i.i.d. p-stable random variables, and the entries of S are those of $\widetilde{S}$ but rescaling by $\Theta(1/t^{1/p})$. Consider the following two cases:

Case 1: $p = 1$: The $\widetilde{S}_{ij}$ are standard Cauchy random variables. Consider the half-Cauchy random variables $X_{i,j} = |S_{i,j}|$. The cumulative distribution function of a half-Cauchy random variable X is $F(x) = \int_0^x \frac{2}{\pi(t^2+1)} dt = 1 - \Theta(\frac{1}{x})$. Thus, for any $i \in [t]$ and $j \in [d]$, $\Pr[|\widetilde{S_{ij}}| \leq D] = 1 - \Theta(\frac{1}{D})$, and $\Pr[|S_{ij}| \leq D] = 1 - \Theta(\frac{1}{tD}) \geq 1 - \Theta(\frac{1}{D})$.

Case 2: $p \in (1, 2)$: We apply the upper tail bound for p-stable random variables in **Lemma** 2.2. For any fixed $i \in [t]$ and $j \in [d]$, $\Pr[|\widetilde{S}_{ij}|^p \leq D^p] \geq 1 - \Theta(\frac{\log D}{D^p})$, which implies that $\Pr[|\widetilde{S_{ij}}| \leq D] \geq 1 - \Theta(\frac{1}{D})$, since $p > 1$. In addition, by the same argument, $\Pr[|S_{ij}| \leq D] = \Pr[|\widetilde{S}_{ij}| \leq t^{1/p} D] = 1 - \Theta(\frac{1}{t^{1/p} D}) \geq 1 - \Theta(\frac{1}{D})$.

Therefore, for $p \in [1, 2)$, if we let $\mathcal{E}_1$ be the event that for all $i \in [t]$ and $j \in [m]$, we simultaneously have $|S_{ij}| \leq D$, then by a union bound over all the entries in S, $\Pr[\mathcal{E}_1] \geq 1 - \Theta(\frac{td}{D})$. In particular, if we choose $D = \text{poly}(nd)$, then $\mathcal{E}_1$ occurs with probability at least $1 - \frac{1}{\text{poly}(nd)}$. Note that if $\mathcal{E}_1$ occurs, then this implies that for all $y \in \mathbb{R}^d$,

$$\|Sy\|_p = \Big(\sum_{i=1}^t \Big|\sum_{j=1}^d S_{ij} y_j\Big|^p\Big)^{1/p} \leq \Big(\sum_{i=1}^t D^p \cdot \Big|\sum_{j=1}^d y_j\Big|^p\Big)^{1/p} \leq D t^{1/p} \|y\|_1 \leq D\text{poly}(d)\|y\|_p$$

Consider the unit ℓ_p ball $B = \{y \in \mathbb{R}^d : \|y\|_p = 1, \exists z \in \mathbb{R}^m \text{ s.t. } y = A_{T,j} z\}$ in the column span of $A_{T,j}$. A subset $\mathcal{N} \subset B$ is a γ-net for B if for all $y \in B$ there exists some $u \in \mathcal{N}$ such that $\|y - u\|_p \leq \gamma$, for some distance $\gamma > 0$. There exists such a net $\mathcal{N}$ for B of size $|\mathcal{N}| = (\frac{1}{\gamma})^{O(m)}$ by a standard greedy construction, since the column span of $A_{T,j}$ has dimension at most $m + 1$. Let us choose γ as follows. First let $K = \text{poly}(nd)$ such that $\|Sy\|_p \leq K\|y\|_p$ (recall that $\|Sy\|_p \leq D\text{poly}(d)\|y\|_p$ if $\mathcal{E}_1$ holds). Then, we choose $\gamma = \frac{1}{m^2 K}$. Thus, $|\mathcal{N}| \leq (m^2 K)^{O(m)} = 2^{O(m \log(nd))}$.

Define the event $\mathcal{E}_2(T, j)$ (here the T, j in parentheses signify that $\mathcal{E}_2(T, j)$ is defined in terms of T and j) as follows: for all $y \in \mathcal{N}$ simultaneously, $\|Sy\|_p \geq \|y\|_p$. (Note that $\mathcal{E}_2(T, j)$ depends on T, j since $\mathcal{N}$ is a net for the column span of $A_{T,j}$.) By applying **Lemma** 2.1, and a union bound over all vectors $y \in \mathcal{N}$, we find that for all $y \in \mathcal{N}$ simultaneously, $\|Sy\|_p \geq \|y\|_p$ with probability at least $1 - \frac{|\mathcal{N}|}{e^t} = 1 - \frac{2^{O(m \log(nd))}}{e^t}$ — in other words, $\mathcal{E}_2(T, j)$ has probability at least $1 - \frac{2^{O(m \log(nd))}}{e^t}$.

Now, consider an arbitrary unit vector $x \in B$. There exists some $y \in \mathcal{N}$ such that $\|x - y\|_p \leq \gamma = \frac{1}{m^2 K}$. If we assume that both $\mathcal{E}_1$ and $\mathcal{E}_2(T, j)$ hold, then the following holds as well:

$$\begin{aligned}
\|Sx\|_p &\geq \|Sy\|_p - \|S(x - y)\|_p && \text{Triangle Inequality} \\
&\geq \|y\|_p - \|S(x - y)\|_p && \text{By event } \mathcal{E}_2(T, j) \\
&\geq \|y\|_p - K\|(x - y)\|_p && \text{Implication of event } \mathcal{E}_1 \\
&\geq \|y\|_p - K\gamma && \text{By } \|x - y\|_p \leq \gamma \\
&= \|y\|_p - O\Big(\frac{1}{m^2}\Big) && \\
&= \|x\|_p - O\Big(\frac{1}{m^2}\Big) && \|x\|_p = \|y\|_p = 1
\end{aligned}$$

For a sufficiently large m, $O(\frac{1}{m^2})$ is at most $\frac{1}{2}$, and thus $\frac{\|x\|_p}{2} = \frac{1}{2} \geq \frac{1}{m^2}$. This implies $\|Sx\|_p \geq \|x\|_p - \frac{\|x\|_p}{2} = \frac{\|x\|_p}{2}$. We can rescale S by a factor of 2 so that $\|Sx\|_p \geq \|x\|_p$.

We have shown that $\|Sy\|_p \geq \|y\|_p$ holds simultaneously for *all* unit vectors y in the column span of $A_{T,j}$, conditioning on $\mathcal{E}_1$ and $\mathcal{E}_2(T, j)$. By linearity, we conclude that $\|Sy\|_p \geq \|y\|_p$ $(1 \leq p < 2)$ holds simultaneously for *all* y in the column span of $A_{T,j}$, conditioning on $\mathcal{E}_1$ and $\mathcal{E}_2(T, j)$.

Step 2: Next, we apply a union bound over all possible subsets $T \subset [n]$ of chosen columns from A and all possible single columns A_{*j} for $j \in [n]$, to argue that $\|S(A_T y - A_{*j})\|_p \geq \|A_T y - A_{*j}\|_p$ holds simultaneously for all $y \in \mathbb{R}^{|T|}$ and all $T \subset [n]$ with $|T| = m = k \cdot \text{poly}(\log k)$ and $j \subset [n]$ with high probability.

In **Step 1**, we showed that $\mathcal{E}_2(T, j)$ fails with probability $\frac{2^{O(m \log(nd))}}{e^t}$, for any fixed T, j. Thus, if we define $\mathcal{E}_{2,all}$ to be the event that $\mathcal{E}_2(T, j)$ holds for all T, j (in other words, $\mathcal{E}_{2,all} = \bigcap_{T,j} \mathcal{E}_2(T, j)$), then the failure probability of $\mathcal{E}_{2,all}$ is at most

$$\frac{2^{O(m \log(nd))}}{e^t} \cdot \binom{n}{m} \cdot d \leq \frac{2^{O(m \log(nd))}}{e^t} \cdot n^{O(m)} \cdot d = \frac{2^{O(m \log(nd))}}{e^t}$$

In summary, if we let $D = \text{poly}(nd)$, then $\mathcal{E}_1$ succeeds with probability $1 - \Theta(\frac{td}{D}) \geq 1 - \frac{1}{\text{poly}(nd)}$. In addition, if we let $D = \text{poly}(nd)$, and let $K = D\text{poly}(d) = \text{poly}(nd)$, then $\mathcal{E}_{2,all}$ holds with probability $1 - \frac{2^{O(m \log(nd)}}{e^t}$. Note that if both $\mathcal{E}_1$ and $\mathcal{E}_{2,all}$ hold, then $\mathcal{E}_1$ and $\mathcal{E}_2(T, j)$ hold for all T, j, meaning that

$$\|A_T y - A_{*j}\|_p \leq \|S(A_T y - A_{*j})\|_p$$

for all $T \subset [n]$ with $|T| = m = k \cdot \text{poly}(\log k)$, $j \in [n]$ and $y \in \mathbb{R}^{|T|}$. Moreover, $\mathcal{E}_1$ and $\mathcal{E}_{2,all}$ simultaneously hold with probability at least $1 - \Theta(\frac{td}{D}) - \frac{2^{O(m \log(nd)}}{e^t}$, which is $1 - \frac{1}{\text{poly}(nd)}$ for $D = \text{poly}(nd)$ and $t = \Theta(m \log(nd))$. This completes the proof of the lemma. □

Lemma 2 (Sketched Error Lower Bound)**.** *Let $A \in \mathbb{R}^{d \times n}$ and $k \in \mathbb{N}$. Let $t = k \cdot poly(\log(nd))$, and let $S \in \mathbb{R}^{t \times d}$ be a matrix whose entries are i.i.d. standard p-stable random variables, rescaled by $\Theta(1/t^{\frac{1}{p}})$. Then, with probability $1 - o(1)$, for* ***all*** *$T \subset [n]$ with $|T| = k \cdot poly(\log k)$ and for all $V \in \mathbb{R}^{|T| \times n}$,*

$$\|A_T V - A\|_p \leq \|SA_T V - SA\|_p$$

Proof. Let y_j denote the j-th column of V, where $j \in [n]$. By applying **Lemma** 2.3, and a union bound over all columns of V, the following holds with probability $1 - \frac{n}{\text{poly}(nd)} = 1 - o(1)$:

$$\begin{aligned} \|A_T V - A\|_p &= (\sum_{j=1}^{n} \|A_T y_j - A_j\|_p^p)^{\frac{1}{p}} \\ &\leq (\sum_{j=1}^{n} \|S(A_T y_j - A_j)\|_p^p)^{\frac{1}{p}} \\ &= \|SA_T V - SA\|_p \end{aligned}$$

□

A.3 Sketched Error Upper Bound (Lemma 3)

We show an upper bound on the approximation error of k-CSS$_p$ on a sketched subset of columns, $\|SA_T V - SA_T\|_p$, which holds for a fixed subset A_T of columns and for the minimizing right factor $V = \arg\min_V \|SA_T V - SA\|_p$ for that subset of columns.

We first adapt Lemma E.17 from [2] to establish an upper bound on the error $\|SA_T V - SA\|_p$ for any fixed V in **Lemma** 3.1. We then apply **Lemma** 3.1 to the minimizer V to conclude the upper bound in **Lemma** 3.

Lemma 3.1. *(An Upper Bound on Norm of A Sketched Matrix) Given $A \in \mathbb{R}^{n\times d}$ and $p \in [1, 2)$, and $U \in \mathbb{R}^{n\times k}$ and $V \in \mathbb{R}^{k\times d}$, if $S \in \mathbb{R}^{t\times n}$ is a dense p-stable matrix, whose entries are rescaled by $\Theta\left(\frac{1}{t^{\frac{1}{p}}}\right)$, then with probability at least $1 - o(1)$,*

$$\|SUV - SA\|_p^p \leq O(\log(td))\|UV - A\|_p^p$$

Here, the failure probability $o(1)$ can be arbitrarily small.

Proof. Lemma E.17 from [2]. □

Lemma 3 (Sketched Error Upper Bound (Lemma E.11 of [2]))**.** *Let $A \in \mathbb{R}^{d\times n}$ and $k \in \mathbb{N}$. Let $t = k \cdot poly(\log(nd))$, and let $S \in \mathbb{R}^{t\times d}$ be a matrix whose entries are i.i.d. standard p-stable random variables, rescaled by $\Theta(1/t^{\frac{1}{p}})$. Then, for a fixed subset $T \subset [n]$ of columns with $|T| = k \cdot poly(\log k)$ and a fixed $V \in \mathbb{R}^{|T|\times n}$, with probability $1 - o(1)$, we have*

$$\min_V \|SA_T V - SA\|_p \leq \min_V O(\log^{1/p}(nd))\|A_T V - A\|_p$$

Proof. Let $X_1^* = \arg\min_X \|SA_T X - SA\|_p$ and $X_2^* = \arg\min_X \|A_T X - A\|_p$. By **Lemma** 3.1,

$$\begin{aligned}\|SA_T X_1^* - SA\|_p^p &\leq \|SA_T X_2^* - SA\|_p^p \\ &\leq O(\log(k\text{poly}(\log n)d))\|A_T X_2^* - A\|_p^p \\ &\leq O(\log(nd))\|A_T X_2^* - A\|_p^p\end{aligned}$$

Therefore,

$$\min_X \|SA_T X - SA\|_p \leq \min_X O(\log^{1/p}(nd))\|A_T X - A\|_p$$

. □

B ℓ_p Lewis Weights and Applications

B.1 ℓ_p Lewis Weights Background

Our streaming and distributed k-CSS algorithms make use of $\ell_{p,2}$ strong coresets and an $O(1)$-approximation k-CSS$_{p,2}$ subroutine, both of which applies importance sampling of the input matrix, based on the so-called *Lewis weights* (see Definition 2), which can be approximated with repeated computation of the *leverage scores* (see Definition 1) in polynomial time [3]. In this section, we briefly introduce the Lewis weights and the desired properties associated with it. We further introduce ℓ_p *sensitivities* and ℓ_p *well-conditioned basis* to aid the analysis of the desired property we need from Lewis weights.

Definition 1 (Statistical Leverage Scores — Definition 16 of [4]). *Let $A \in \mathbb{R}^{n\times d}$, and suppose $A = U\Sigma V^T$ is the "thin" SVD of A.* [1] *Then, for $i \in [n]$, define $\ell_i(A) = \|U_{i,*}\|_2^2$ — we say $\ell_i(A)$ is the i^{th} statistical leverage score of A.*

Definition 2 (ℓ_p Lewis Weights — Definition 2.2 of [3]). *Let $1 \le p < \infty$, and let $A \in \mathbb{R}^{n\times d}$. Then, the ℓ_p Lewis weights of A are given by a unique vector $\overline{w} \in \mathbb{R}^n$ such that $\overline{w}_i = \ell_i(diag(\overline{w})^{1/2-1/p}A)$, where $diag(\overline{w})$ is the $n \times n$ diagonal matrix with the entries of $\overline{w}$ on its diagonal. By Corollaries 3.4 and 4.2 of [3], such a vector $\overline{w}$ exists and is unique.*

Definition 3 (ℓ_p Sensitivities [5]). *Let $1 \le p < \infty$, and let $A \in \mathbb{R}^{n\times d}$. Let $col(A)$ denote the column span of A. The i^{th} ℓ_p sensitivity of A is defined as $\sup_{y\in col(A)} \frac{y_i^p}{\|y\|_p^p}$.*

Definition 4 (ℓ_p Well-conditioned Basis — Definition 3 of [6]). *Let $1 \le p < \infty$ and $A \in \mathbb{R}^{n\times d}$ of rank k. Let q be its dual norm. Then an $n \times k$ matrix U is an (α, β, p) well-conditioned basis for the column span of A, if (1) $\|U\|_p \le \alpha$, and (2) for all $z \in \mathbb{R}^k$, $\|z\|_q \le \beta\|Uz\|_p$. W will say U is a p well-conditioned basis for the column span of A, if α and β are $k^{O(1)}$, independent of d and n.*

Definition 5 (ℓ_p Leverage score sampling — Theorem 5 of [7]). *Let $1 \le p < \infty$ and $A \in \mathbb{R}^{n\times d}$ of rank k. Let U be an (α, β, p) well-conditioned basis for the column span of A. Given approximation error ε and failure probability δ, for $r \ge C(\varepsilon, \delta, p, k)$, the ℓ_p leverage score sampling is any sampling probability $p_i \ge \min\{1, \frac{\|U_{i*}\|_p^p}{\|U\|_p^p} r\}$, $\forall i \in [n]$.*

The desired property from Lewis weights we need is called an ℓ_p subspace embedding (see **Theorem** 4.2). [3] shows for a matrix $A \in \mathbb{R}^{n\times d}$, if the rows of A are appropriately sampled using a certain distribution based on the ℓ_p Lewis weights of A, this property holds with constant probability. However, for our construction of strong coresets, we need this property of Lewis weights to hold with high probability $1 - \delta$ for some small $\delta \in (0, \frac{1}{2})$. We explain why this is possible following the works from [3, 8].

Theorem 4.2. *(ℓ_p-Lewis Weights Subspace Embedding) Given an input matrix $A \in \mathbb{R}^{n\times d}$ and $p \in [1, 2)$, there exists a distribution $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ on the rows of A, where the distribution is constructed based on Lewis weights sampling. If the following two conditions are met: (1) $n \le poly(d/\varepsilon)$, and (2) the minimum (row) Lewis weights of A is at least $1/poly(d/\varepsilon)$, then for a sampling and rescaling matrix S with t rows, each chosen independently as the i^{th} standard basis vector times $\frac{1}{(t\lambda_i)^{\frac{1}{p}}}$ with probability λ_i, with $t = O(d \cdot poly(\log(d/\delta), 1/\varepsilon))$, the following holds for all $x \in \mathbb{R}^d$ simultaneously with probability $1 - \delta$:*

$$\|SAx\|_p = (1 \pm \epsilon)\|Ax\|_p$$

Proof. The Theorem follows Theorem 7.1 of [3], except that [3] states the above property of Lewis weights holds with constant probability. However, this result can be improved for it to hold with probability $1-\delta$ as follows: Using the results of [8], it is possible to construct a sampling and rescaling matrix S (i.e. a matrix with one non-zero value per row) with $d$poly$(\log(d/\delta), 1/\varepsilon)$ rows such that with probability at least $1 - \delta$, we have $\|SAx\|_p = (1 \pm \varepsilon)\|Ax\|_p$ simultaneously $\forall x \in \mathbb{R}^d$.

To do this, the authors of [8] construct a sequence of $v = \text{poly}((\log d)/\varepsilon)$ sets of vectors $\{V_i\}_{i=1}^v$, and each vector in a net over the column space can be written approximately as a sum of vectors, one drawn from each set V_i. Then they show via Bernstein's inequality that the vectors in all sets have their norms preserved if one samples from the ℓ_p-Lewis weights of A, and the final bound follows from the triangle inequality. By increasing the number of rows in S by an $O(\log(d/(\varepsilon\delta))$ factor,

[1] meaning that if A is of rank k, then U and V have k columns, and $\Sigma \in \mathbb{R}^{k\times k}$.

one can now argue that with probability $1-\delta$, all vectors y in the column span of A have their norm $\|y\|_p$ preserved. This gives a total of $O(d\text{poly}(\log(d/\delta), 1/\varepsilon))$ rows in S. However, to apply the results from [8], we need two conditions to be satisfied: (1) $n \le \text{poly}(d/\varepsilon)$, and (2) the minimum (row) Lewis weight of A is at least $1/\text{poly}(d/\varepsilon)$.

We can achieve both conditions by first replacing A with TA, where T is a sampling matrix for which T has $\text{poly}(d/\varepsilon)$ rows and with probability $1-\delta$, $\|TAx\|_p = (1 \pm \varepsilon)\|Ax\|_p$ simultaneously $\forall x \in \mathbb{R}^d$. Many constructions of such T exist with $\text{poly}(d/\varepsilon \log(1/\delta))$ rows, e.g. e.g., based on the ℓ_p-sensitivities of A (see Definition 3) or ℓ_p leverage score sampling (see Definition 5), or the ℓ_p-Lewis weights themselves (see Definition 2). See for example Theorem 10 of [5]. If we choose $\delta > 1/\exp(\text{poly}(d/\varepsilon))$, then the number of rows of TA will be at most $\text{poly}(d/\varepsilon)$, satisfying condition (1) of Theorem 4.2. On the other hand, if $\delta \le 1/\exp(\text{poly}(d/\varepsilon))$, then $\log(1/\delta) > \text{poly}(d/\varepsilon)$. In this case, we can just sample using ℓ_p sensitivities as in Theorem 3.10 of [5]. The number of rows needed will be $\text{poly}(d/\varepsilon)$, which can simply be absorbed into the $\text{poly}(\log(1/\delta))$.

While (1) holds since the number of rows of T is at most $\text{poly}(d/\varepsilon)$, we can ensure (2) also holds by computing the (row) ℓ_p-Lewis weights of TA, and discarding any row i with ℓ_p-Lewis weight less than $1/\text{poly}(n/\varepsilon)$. By Lemma 5.5 of [3] this cannot make the ℓ_p-Lewis weight of any non-discarded row decrease. Moreover, since the ℓ_p-Lewis weights are upper bounds on the ℓ_p-sensitivities for $1 < p < 2$ (by Lemma 3.8 of [5] and Definition 3), discarding such rows i only changes $\|TAx\|_p$ by a $(1 \pm \varepsilon)$ factor for any $x \in \mathbb{R}^d$, by the triangle inequality. Finally, since $n \le \text{poly}(d/\varepsilon)$, we also have that any ℓ_p-Lewis weight is now at least $1/\text{poly}(d/\varepsilon)$, as needed to now apply the result of [8]. Thus, we can now apply the above S to non-discarded rows of TA. □

B.2 Strong Coresets for $\ell_{p,2}$ Norm Low Rank Approximation (Lemma 4)

Lemma 4 (Strong Coresets in $\ell_{p,2}$ norm [9]). *Let $A \in \mathbb{R}^{d\times n}$, $k \in \mathbb{N}$, $p \in [1, 2)$, and $\varepsilon, \delta \in (0, 1)$. Then, in $\widetilde{O}(nd)$ time, one can find a sampling and reweighting matrix T with $O(\frac{d}{\varepsilon^2} poly(\log(d/\varepsilon), \log(1/\delta)))$ columns, such that, with probability $1-\delta$, for all rank-k matrices U,*

$$\min_{\textit{rank-}k\ V} \|UV - AT\|_{p,2} = (1 \pm \varepsilon) \min_{\textit{rank-}k\ V} \|UV - A\|_{p,2}$$

*where AT is called a **strong coreset** of A.*

Proof. We can obtain T with the desired number of columns using the strong coreset construction from **Lemma 16** in [9]. For our purposes, the matrix $B \in \mathbb{R}^{n\times(d+1)}$ that we use will be different than the B used in the statement and proof of **Lemma 16** in [9]. The coreset construction in [9] has the goal of removing a dependence on d in the coreset size. In [9], B refers to a matrix obtained by projecting A onto a $\text{poly}(k)$-dimensional subspace S obtained by running a dimensionality reduction algorithm (referred to as DimensionalityReduction in [9]) and constructing a coreset by sampling rows from B. The rows are sampled according to the ℓ_p Lewis weights of B.

In our case, we do not want our coreset size to have a polynomial dependence on k, while a linear dependence on d suffices. Thus, instead of using the dimensionality reduction subroutine in [9], we simply let B be the input matrix A, concatenated with a column of 0's (the column span of A will be the subspace S referred to in the statement of **Lemma 16** of [9]). The desired number of rows and running time then follows from **Lemma 16** of [9]. [2] Based on $1-\delta$ ℓ_p Lewis weights subspace embedding, the size of the coreset grows linearly in $\text{poly}(\log(1/\delta))$. □

B.3 Bi-criteria $O(1)$-approximation algorithm for k-CSS$_{p,2}$ (Theorem 1)

We introduce an $O(1)$-approximate bi-criteria k-CSS$_{p,2}$ algorithm (**Algorithm** 1), which is a modification of the algorithm from [10]. The major difference is that we use ℓ_p-Lewis weight sampling, instead of ℓ_p leverage score sampling, which reduces the number of output columns from $O(k^2)$ to $O(k\text{poly}(\log k))$.

We first show how to use a sparse embedding matrix S to obtain an $O(1)$-approximate left factor in Section B.3.1. We then show how to apply the ℓ_p-Lewis weight sampling to select a subset of $\widetilde{O}(k)$ columns that gives an $O(1)$-approximation in Section B.3.2. Finally, we conclude the analysis of our $O(1)$-approximate bi-criteria k-CSS$_{p,2}$ algorithm in Section B.3.3.

[2] The proof of **Lemma 16** of [9] mentions that for $p > 1$, Lewis weight sampling requires $(f/\varepsilon)^{O(p)}$ rows for a matrix with C columns — this is a typo, and $f\text{poly}(\frac{\log f}{\varepsilon})$ rows suffice.

Algorithm 1 polynomial time, $O(1)$-approximation for k-$\text{CSS}_{p,2}$ $(1 \le p < 2)$

```
**Input:** The data matrix $A \in \mathbb{R}^{d\times n}$, rank $k \in \mathbb{N}$
**Output:** The left factor $U \in \mathbb{R}^{d\times\widetilde{O}(k)}$, the right factor $V \in \mathbb{R}^{\widetilde{O}(k)\times n}$ such that $\|UV - A\|_{p,2} \le O(1)\min_{\text{rank-k}A_k}\|A_k - A\|_{p,2}$
$S \leftarrow \widetilde{O}(k) \times d$ sparse embedding matrix, with sparsity $s = \text{poly}(\log k)$.
$S' \leftarrow n \times \widetilde{O}(k)$ sampling matrix, each column of which is a standard basis vector chosen randomly according to the $\ell_p$ Lewis weights of columns of $SA$.
Return $U \leftarrow AS'$, $V \leftarrow (AS')^{\dagger}A$ {† denotes the Moore-Penrose pseudoinverse.}
```

B.3.1 Sparse Embedding Matrices

The **sparse embedding matrix** $S \in \mathbb{R}^{\widetilde{O}(k)\times d}$ of [11], and used by [10], is constructed as follows: each column of S has exactly s non-zero entries chosen in uniformly random locations. Each non-zero entry is a random value $\pm\frac{1}{\sqrt{s}}$ with equal probability. s is also called the *sparsity* of S. Let h be the hash function that picks the location of the non-zero entries in each column of S and σ be the hash function that determines the sign $\pm$ of each non-zero entry.

Applying the sparse embedding matrix S to A enables us to obtain a rank-k right factor that is at most a factor of $O(1)$ worse than the best rank-k approximation error in the $\ell_{p,2}$ norm. We adapt Theorem 32 from [10] to show this in **Theorem** 5.5. Notice that in Theorem 32 of [10], the number of rows required for S is $O(k^2)$, but this can be reduced to $\widetilde{O}(k)$ through a different choice of hyperparameters when constructing the sparse embedding matrix S.

We note two choices of hyperparameters, i.e., the number m of rows and sparsity s, of S in **Theorem** 5.1 and **Theorem** 5.2, both of which give the same result. The proof of Theorem 32 from [10] uses the hyperparameters from **Theorem** 5.1. We instead use the hyperparameters from **Theorem** 5.2 and show in **Lemma** 5.3 that $\widetilde{O}(k)$ rows of S suffice to preserve certain desired properties. We then combine **Lemma** 5.3 and **Lemma** 5.4 adapted from [10], to conclude our result in **Theorem** 5.5, following the analysis from [10].

Theorem 5.1. *(Theorem 3 from [11]) For a sparse embedding matrix $S \in \mathbb{R}^{m\times n}$ with sparsity $s = 1$ and a data matrix $U \in \mathbb{R}^{n\times d}$, let $\epsilon \in (0,1)$. With probability at least $1-\delta$ all singular values of SU are $(1\pm\epsilon)$ as long as $m \ge \delta^{-1}(d^2+d)/(2\epsilon-\epsilon^2)^2$. For the hash functions used to construct S, σ is 4-wise independent and h is pairwise independent.*

Theorem 5.2. *(Theorem 9 from [11]) For a sparse embedding matrix $S \in \mathbb{R}^{m\times n}$ with sparsity $s = \Theta(\log^3(d/\delta)/\epsilon)$ and a data matrix $U \in \mathbb{R}^{n\times d}$, let $\epsilon \in (0,1)$. With probability at least $1-\delta$ all singular values of SU are $(1\pm\epsilon)$ as long as $m = \Omega(d\log^8(d/\delta)/\epsilon^2)$. For the hash functions used to construct S, we have that σ, h are both $\Omega(\log(d/\delta))$-wise independent.*

Lemma 5.3. *Let $\mathcal{C}$ be a constraint set and $A \in \mathbb{R}^{n\times d}, B \in \mathbb{R}^{n\times d'}$ be two arbitrary matrices. For a sparse embedding matrix $S \in \mathbb{R}^{m\times n}$, there is $m = O(\frac{d\log^8(\frac{d}{\epsilon^{p+1}})}{\epsilon^{2(p+1)}})$, such that with constant probability, the following hold:*

i) $\|S(AX-B)\|_{p,2} \ge (1-\epsilon)\|AX-B\|_{p,2}$ *for all* $X \in \mathbb{R}^{d\times d'}$

ii) $\|S(AX^*-B)\|_{p,2} \le (1+\epsilon)\|AX^*-B\|_{p,2}$, *where* $X^* = \arg\min_{X\in\mathcal{C}}\|AX-B\|_{p,2}$

Proof. The proof is the same as the proof of Lemma 29 from [10], except that we use a different choice of hyperparameters in constructing S, i.e., sparsity s and the number m of rows. In the proof of Lemma 29 from [10], the construction of S follows **Theorem** 5.1, where the sparsity $s = 1$, but requires $m = O(d^2)$ rows. We replace the construction by **Theorem** 5.2, where we pick $\delta = \epsilon^{p+1}$. Now the sparsity s is larger but this construction reduces the number of rows required to $m = \widetilde{O}(d)$.

If we use the construction in Theorem 5.2 with the parameters ε, δ both being ε^{p+1}, then the rest of the proof follows from Lemma 27 of [10] (using the same argument as in Lemma 29 of [10]). As in Lemma 29 of [10], properties (i) and (ii) of Lemma 27 of [10] follow simply because we have chosen the parameters ε, δ of S to be ε^{p+1} (thus, by Theorem 5.2, S is an ε-subspace embedding for A in the ℓ_2 norm, and S is an ε^{p+1}-subspace embedding for $[A, B_{*,i}]$ with probability $1-\varepsilon^{p+1}$, for all i). Finally, to show that property (iii) in Lemma 27 of [10], the only property of the matrix S that is needed by [10] is Equation (20) of [11], which also holds when S is constructed as in Theorem 9 of [11]. □

Lemma 5.4. *Consider a data matrix $A \in \mathbb{R}^{n \times d}$. Let the best rank-k matrix in the $\ell_{p,2}$ norm be $A_k = \arg\min_{\textit{rank-k } A_k} \|A_k - A\|_{p,2}$. For $R \in \mathbb{R}^{d \times m}$, if R^T satisfies both of the following two conditions for all $X \in \mathbb{R}^{n \times n}$:*

i) $\|R^T(A_k^T X - A^T)\|_{p,2} \geq (1-\epsilon)\|A_k^T X - A^T\|_{p,2}$

ii) $\|R^T(A_k^T X^* - A^T)\|_{p,2} \leq (1+\epsilon)\|A_k^T X^* - A^T\|_{p,2}$, *where* $X^* = \arg\min_X \|A_k^T X - A^T\|_{p,2}$

then

$$\min_{\textit{rank-k } X} \|XR^T A_k^T - A^T\|_{p,2}^p \leq (1+3\epsilon)\|A_k^T - A^T\|_{p,2}^p$$

Proof. Lemma 31 from [10]. □

Theorem 5.5. *($\ell_{p,2}$-Low Rank Approximation) Let the data matrix be $A \in \mathbb{R}^{d \times n}$ and $k \in \mathbb{N}$ be the desired rank. Let $S \in \mathbb{R}^{m \times d}$ be a sparse embedding matrix with $m = O(k\mathrm{poly}(\log k)\mathrm{poly}(\frac{1}{\varepsilon}))$ rows, and sparsity $s = \mathrm{poly}(\log k)$. Then, the following holds with constant probability:*

$$\min_{\textit{rank-k } X} \|XSA - A\|_{p,2} \leq (1+3\varepsilon) \min_{\textit{rank-k } A_k} \|A_k - A\|_{p,2}$$

Proof. The proof is the same as the proof of Theorem 32 in [10], except that we adapt a different construction of the sparse embedding matrix S, which reduces the number of rows from $O(k^2)$ to $\widetilde{O}(k)$ with increased sparsity s.

Consider $A_k = \arg\min_{\text{rank-k } A_k} \|A_k - A\|_{p,2}$. Let V_k be a basis for the column space of A_k. By applying **Lemma** 5.3 and **Lemma** 5.4 on the basis V_k, we conclude the above theorem by setting the number m of rows to $m = O\Big(\frac{k \log^8(\frac{k}{\varepsilon^2})}{\varepsilon^4}\Big)$, and sparsity $s = \mathrm{poly}(\log k)$ in the sparse embedding matrix S. □

B.3.2 Using Lewis Weight Sampling for Column Subset Selection

Here we show how to use ℓ_p Lewis weights sampling (discussed above in at the beginning of Section B) for k-CSS$_{p,2}$. We first introduce a technical tool, a version of Dvoretzky's Theorem (**Theorem** 5.7) which allows us to embed ℓ_2^n into $\ell_2^{O(n/\varepsilon^2)}$ with only $(1 \pm \varepsilon)$ distortion, and thus enables us to switch between the ℓ_p norm and the ℓ_2 norm and use ℓ_p Lewis weight sampling. Based on **Theorem** 4.2 and **Theorem** 5.7, we show in **Theorem** 5.8 that Lewis weight sampling provides a good subset of columns, on which our later analysis of k-CSS$_{p,2}$ is based.

Theorem 5.7. *(Randomized Dvoretzky's Theorem) Let $n \in \mathbb{N}$, and $\varepsilon \in (0,1)$. Let $r = \frac{n}{\varepsilon^2}$. Let $G \in \mathbb{R}^{r \times n}$ be a random matrix whose entries are i.i.d. standard Gaussian random variables, rescaled by $\frac{1}{\sqrt{r}}$. For $r = \frac{n}{\epsilon^2}$, the following holds with probability $1 - e^{-\Theta(n)}$, for all $y \in \mathbb{R}^n$,*

$$\|Gy\|_p = (1 \pm \varepsilon)\|y\|_2$$

Proof. This follows from Theorem 1.2 from [12]. □

Theorem 5.8. *(Subset of Columns by Lewis Weights Sampling) Let $A \in \mathbb{R}^{d \times n}$. Let $S \in \mathbb{R}^{m \times d}$ be a sparse embedding matrix, with $m = O(k \cdot \mathrm{poly}(\log k)\mathrm{poly}(\frac{1}{\epsilon}))$. Further, let $S' \in \mathbb{R}^{n \times t}$ be a sampling matrix whose columns are random standard basis vectors generated according to the ℓ_p Lewis weights of columns of SA (that is, the row sampling matrix $(S')^T$ is generated based on the Lewis weights of $(SA)^T$), with $t = k \cdot \mathrm{poly}(\log k)$. Then, for $\hat{X} = \arg\min_{\textit{rank-k } X} \|XSAS' - AS'\|_{p,2}$, the following holds with probability $1 - o(1)$:*

$$\|\hat{X}SA - A\|_{p,2} \leq \Theta(1) \min_{\textit{rank-k } A_k} \|A_k - A\|_{p,2}$$

Proof. Let $X^* = \arg\min_{\text{rank-k } X^*} \|X^*SA - A\|_{p,2}$. By the triangle inequality,

$$\|\hat{X}SA - A\|_{p,2} \leq \|X^*SA - \hat{X}SA\|_{p,2} + \|X^*SA - A\|_{p,2}$$

Our goal is to bound $\|X^*SA - \hat{X}SA\|_{p,2}$. By Lemma D.28 and Lemma D.29 from [2], for any column sampling matrix S and for any fixed matrix Y, it can be shown that $\mathbf{E}[\|YS\|_p^p] = \|YS\|_p^p$. In our case, since S' is a sampling matrix, we have $\mathbf{E}[\|YS'\|_p^p] = \|YS'\|_p^p$ for any fixed matrix Y.

Now let $G \in \mathbb{R}^{\Theta(d)\times d}$ be a rescaled random matrix whose entries are i.i.d. standard Gaussian random variables as in **Theorem** 5.7. We apply **Theorem** 5.7 to transform between the ℓ_p space and the Euclidean space. Since transformation of both directions can be done with very small distortion, we obtain a $\Theta(1)$ approximation. With constant probability, we have

$$\begin{aligned}
\|X^*SA - \hat{X}SA\|_{p,2} &= \Theta(1)\|G(X^* - \hat{X})SA\|_p && \text{By } \textbf{Theorem } 5.7\\
&= \Theta(1)\|G(X^* - \hat{X})SAS'\|_p && \text{By } \textbf{Theorem } 4.2\\
&= \Theta(1)\|(X^* - \hat{X})SAS'\|_{p,2} && \text{By } \textbf{Theorem } 5.7\\
&\le \Theta(1)\Big(\|X^*SAS' - AS'\|_{p,2} + \|\hat{X}SAS' - AS'\|_{p,2}\Big) && \text{Triangle Inequality}\\
&\le \Theta(1)\|X^*SAS' - AS'\|_{p,2} && \text{Since } \hat{X} = \arg\min_{\text{rank-k } X} \|XSAS' - AS'\|_{p,2}\\
&= \Theta(1)\|G(X^*SA - A)S'\|_p && \text{By } \textbf{Theorem } 5.7\\
&\le \Theta(1)\|G(X^*SA - A)\|_p && \text{By Markov Bound on } \mathbf{E}[\|YS'\|_p^p] = \|YS'\|\\
&= \Theta(1)\|X^*SA - A\|_{p,2} && \text{By } \textbf{Theorem } 5.7
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\hat{X}SA - A\|_{p,2} &\le \|X^*SA - \hat{X}SA\|_{p,2} + \|X^*SA - A\|_{p,2}\\
&\le \Theta(1)\|X^*SA - A\|_{p,2}\\
&\le \Theta(1) \min_{\text{rank-k } A_k} \|A - A_k\|_{p,2} && \text{By } \textbf{Theorem } 5.5
\end{aligned}$$

as desired. Note that we can achieve $o(1)$ failure probability by increasing the number of columns in S' by a logarithmic factor — see **Theorem** 4.2. □

B.3.3 Analysis for k-CSS$_{p,2}$

We now conclude our proof that Algorithm 1 for bi-criteria k-CSS$_{p,2}$ achieves an $O(1)$ approximation factor with polynomial running time. This result is stated as **Theorem** 1.

Theorem 1 (Bicriteria $O(1)$-Approximation Algorithm for k-CSS$_{p,2}$)**.** *Let $A \in \mathbb{R}^{d\times n}$ and $k \in \mathbb{N}$. There is an algorithm with $(nnz(A) + d^2)\cdot k poly(\log k)$ runtime that outputs a rescaled subset of columns $U \in \mathbb{R}^{d\times\widetilde{O}(k)}$ of A and a right factor $V \in \mathbb{R}^{\widetilde{O}(k)\times n}$ for which $V = \min_V \|UV - A\|_{p,2}$, such that with probability $1 - o(1)$,*

$$\|UV - A\|_{p,2} \le O(1)\cdot \min_{\textit{rank-k } A_k} \|A_k - A\|_{p,2}$$

Proof. **Approximation Factor.** First notice that the minimizer $\hat{X}$ of $\|\hat{X}SAS' - AS'\|_{p,2}$ has to be in the column span of AS'. Thus we can write $\hat{X} = (AS')Y$ for some matrix Y. By **Theorem** 5.8,

$$\|\hat{X}SA - A\|_{p,2} = \|(AS')YSA - A\|_{p,2} \le \Theta(1) \min_{\text{rank-k } A_k} \|A - A_k\|_{p,2}$$

We denote $YSA = V$. We take the left factor $U = AS'$ and solving for $\min_V \|UV - A\|_{p,2}$ will give us a $\Theta(1)$ approximation to $\min_{\text{rank-k } A_k} \|A - A_k\|_{p,2}$. A good minimizer for the right factor V in the Euclidean space is $V = (AS')^\dagger A$. This concludes our result. Notice that since S' is a sampling matrix with $\widetilde{O}(k)$ columns, we get a rank-k left factor U as a subset of columns of A as desired.

Running time. First notice that S is a sparse embedding matrix with poly$(\log k)$ non-zero entries. Thus computing SA takes time nnz$(A)\cdot$ poly$(\log k)$. By [3], computing the Lewis weights of SA takes time nnz(SA) + poly$(k) \le$ nnz(A)poly$(\log k)$ + poly(k), and computing the output left factor $U = AS'$ takes time nnz(A). Computing $(AS')^\dagger$ takes time $d^2 \cdot k$poly$(\log k)$. Computing the right factor $V = (AS')^\dagger A$ takes nnz$(A)k$poly$(\log k)$. Therefore, the overall running time is $(\text{nnz}(A) + d^2)\cdot k\text{poly}(\log k)$.

Failure Probability. For the failure probability of the first step, note that we can select the parameter δ for the sparse embedding matrix S to be $\frac{1}{\text{poly}(d)}$, at the cost of a logarithmic factor in the number of rows. Similarly, the failure probability of the second step is $o(1)$ as mentioned in Theorem 5.8. □

C The Streaming Algorithm and Full Analysis (Section 4)

We give a single-pass streaming algorithm for k-CSS$_p$ ($1 \leq p < 2$) in **Algorithm** 2, which is based on the Merge-and-Reduce framework (see, e.g. [13]) previously used in graph streaming algorithms, for instance.

Algorithm 2 A one-pass streaming algorithm for bi-criteria k-CSS$_p$ in the column-update streaming model.

```
**Input:** A matrix $A \in \mathbb{R}^{d\times n}$ whose columns arrive one at a time, $p \in [1,2)$, rank $k \in \mathbb{N}$ and batch size $r = k\text{poly}(\log(nd))$.
**Output:** A subset of $\widetilde{O}(k)$ columns $A_I$.
Generate a dense $p$-stable sketching matrix $S \in \mathbb{R}^{k\text{poly}(\log(nd))\times d}$.
A list $C \leftarrow \{\}$ of strong coresets and corresponding level numbers.
A list $D \leftarrow \{\}$ of unsketched column subsets of $A$ corresponding to the list $C$ of strong coresets and their level numbers.
A list of sketched columns $M \leftarrow \{\}$.
A list of corresponding unsketched columns $L \leftarrow \{\}$.
for Each column $A_{*j}$ seen in the data stream do
  $M \leftarrow M \cup SA_{*j}$
  $L \leftarrow L \cup A_{*j}$
  if length of $M == r$ then
    $C \leftarrow C \cup (M, 0), D \leftarrow D \cup L$
    $C, D \leftarrow$ Recursive Merge$(C, D)$ {// Algorithm 3}
    $M \leftarrow \{\}, L \leftarrow \{\}$
  end if
end for
$C \leftarrow C \cup (M, 0), D \leftarrow D \cup L$
$C, D \leftarrow$ Recursive Merge$(C, D)$ {// Algorithm 3}
$I \leftarrow$ The column indices obtained by applying to the concatenation of the strong coresets in $C$. (Here, $I$ is a set of column indices and $|I| = k \cdot \text{poly}(\log k)$.)
Finally, recover $A_I$ by mapping the selected indices $I$ to unsketched columns in $D$.
```

Algorithm 3 Recursive Merge

```
**Input:** A list $C$ of strong coresets and their corresponding level numbers. A list $D$ of (unsketched) column subsets of $A$ corresponding to the sketched columns in $C$.
**Output:** New $C$, where the list of strong coresets is greedily merged, and the corresponding new $D$.
if length of $C == 1$ then
  Return $C, D$.
else
  Let $(C_{-2}, l_{-2}), (C_{-1}, l_{-1})$ be the second to last and last elements of the list $C$ (i.e. the second to last and last sets of columns $C_{-2}, C_{-1}$ with their corresponding level $l_{-2}, l_{-1}$ from list $C$).
  if $l_{-2} == l_{-1}$ then
    Remove $(C_{-2}, l_{-2}), (C_{-1}, l_{-1})$ from $C$.
    Remove the corresponding $D_{-2}, D_{-1}$ from $D$.
    Compute a strong coreset $C_0$ of (i.e., sample and rescale columns from) $C_{-2} \cup C_{-1}$, as described in the proof of Lemma 4 — $C_0$ has at least $k \cdot \text{poly}(\log nd)$ columns. Record the original indices $I$ in $C_{-2} \cup C_{-1}$ of the columns selected in $C_0$.
    Map indices $I$ to columns in $D_{-2} \cup D_{-1}$ to form a new subset of columns $D_0$.
    $C \leftarrow C \cup (C_0, l_{-1} + 1), D \leftarrow D \cup D_0$.
    Recursive Merge$(C, D)$.
  else
    Return $C, D$.
  end if
end if
```

To analyze our streaming algorithm, we first need **Lemma** 5 to show how the approximation error propagates through each level of the binary tree induced by the merge operator. It shows how a strong coreset C_0, computed at level l of the tree, approximates the projection error for the union of all columns at the leaves of the subtree rooted at C_0.

Lemma 5 (Approximation Error from Merging)**.** *Let C_0 be a strong coreset constructed in a step of Algorithm 3 (**Recursive Merge**), i.e. C_{-1} and C_{-2} are two strong coresets at levels $l-1$ of the binary tree, and C_0 is a strong coreset at level l obtained by taking a strong coreset for the concatenation of C_{-1} and C_{-2}. Then,*

- *If C_0 is a strong coreset of $C_{-1} \cup C_{-2}$, constructed as described in Lemma 4, with at least $\frac{k}{\gamma^2} \cdot poly(\log(nd/\gamma))$ columns, then with probability at least $1 - \frac{1}{n^2}$,*

$$\min_V \|UV - C_0\|_{p,2} = (1 \pm \gamma) \min_V \|UV - (C_{-1} \cup C_{-2})\|_{p,2}$$

  for all matrices U of rank at most k, simultaneously.

- *If $\mathbf{C}$ is a strong coreset at level l of the binary tree, and M is the union of the columns of SA represented as leaves of the subtree rooted at $\mathbf{C}$, then with probability at least $1 - \frac{q}{n^2}$,*

$$\min_V \|UV - \mathbf{C}\|_{p,2} = (1 \pm \gamma)^l \|UV - M\|_{p,2}$$

  for all rank-k matrices U, simultaneously, as long as the coresets computed in each merge operation have at least $\frac{k}{\gamma^2} \cdot poly(\log(nd/\gamma))$ columns. Here, q is the number of nodes in the subtree rooted at $\mathbf{C}$.

Proof. The first statement is a direct consequence of Lemma 4 — we are applying Lemma 4, setting approximation error $\varepsilon = \gamma$, failure probability $\delta = \frac{1}{n^2}$ and rank $k = k$. Note that the coreset construction described in the proof of Lemma 4 requires $O(d \log d/\varepsilon^2) \cdot \log(1/\delta)$ columns to be sampled, but in our case, $d = k \cdot \text{poly}(\log nd)$, since SA has $k \cdot \text{poly}(\log nd)$ rows.

We show the second statement by using the first statement, together with induction on the number of merge operations l, and a union bound over all the merge operations performed. The union bound is simply as follows: note that for each node C_0 in the subtree rooted at $\mathbf{C}$, with probability $1 - \frac{1}{n^2}$,

$$\min_V \|UV - C_0\|_{p,2} = (1 \pm \gamma) \min_V \|UV - (C_{-1} \cup C_{-2})\|_{p,2}$$

for all U of rank k, where C_{-1} and C_{-2} are the two coresets corresponding to the children of C_0 (this is simply by the first statement of Lemma 5). Thus, by a union bound, this holds simultaneously for all nodes C_0 in the subtree rooted at $\mathbf{C}$, with probability at least $1 - \frac{q}{n^2}$, where q is the number of nodes in the subtree rooted at $\mathbf{C}$. In other words, let $\mathcal{E}$ be the event that for all nodes C_0, C_0 is a strong coreset of $C_{-1} \cup C_{-2}$ — then $\mathcal{E}$ occurs with probability at least $1 - \frac{q}{n^2}$. Note that since $q \leq 2n$, a failure probability $\delta = \frac{1}{n^2}$ suffices to pay for the union bound.

Assuming $\mathcal{E}$ holds, we can apply induction. First let us consider the base case where $l = 1$ — here, the desired result clearly holds since it is implied by the event $\mathcal{E}$ (since the subtree rooted at $\mathcal{C}$ only has two other nodes, C_{-1} and C_{-2}).

Now suppose $l > 1$, and the second statement of Lemma 5 holds for $\mathbf{C}$ at levels less than l. Now, suppose $\mathbf{C}$ is at level l, and let C_{-1} and C_{-2} be the coresets corresponding to the children of $\mathbf{C}$ in the binary tree. Let M_{-1}, M_{-2} each be the contiguous submatrices of SA represented by the leaves of the subtrees rooted at C_{-1} and C_{-2} respectively. Note that by its definition, $M = M_{-1} \cup M_{-2}$, where M is as defined in the second statement of Lemma 5. Let q_1, q_2 be the number of nodes in the subtree rooted at C_{-1} and C_{-2} respectively, and $q = q_1 + q_2$ be the number of nodes in the subtree rooted at $\mathbf{C}$. By the induction hypothesis, since C_{-1} and C_{-2} are at levels $l-1$, for all rank-k matrices U, with probability $1 - \frac{q_1}{n^2}$,

$$\min_V \|UV - C_{-1}\|_{p,2} = (1 \pm \gamma)^{l-1} \min_V \|UV - M_{-1}\|_{p,2}$$

and with probability $1 - \frac{q_2}{n^2}$,

$$\min_V \|UV - C_{-2}\|_{p,2} = (1 \pm \gamma)^{l-1} \min_V \|UV - M_{-2}\|_{p,2}$$

Thus, for any matrix U of rank k, with probability $1 - \frac{q}{n^2}$,

$\min_V \|UV - \mathbf{C}\|_{p,2} = (1 \pm \gamma) \min_V \|UV - (C_{-1} \cup C_{-2})\|_{p,2}$

$$
\begin{aligned}
&= (1 \pm \gamma)\Big(\min_V \|UV - C_{-1}\|_{p,2}^p + \min_V \|UV - C_{-2}\|_{p,2}^p \Big)^{1/p} \\
&= (1 \pm \gamma) \cdot \Big((1 \pm \gamma)^{p(l-1)} \cdot \min_V \|UV - M_{-1}\|_{p,2}^p + (1 \pm \gamma)^{p(l-1)} \cdot \min_V \|UV - M_{-2}\|_{p,2}^p \Big)^{1/p} \\
&= (1 \pm \gamma)^l \cdot \Big(\min_V \|UV - M_{-1}\|_{p,2}^p + \min_V \|UV - M_{-2}\|_{p,2}^p \Big)^{1/p} \\
&= (1 \pm \gamma)^l \min_V \|UV - M\|_{p,2}
\end{aligned}
$$

Here, the first equality is because the event $\mathcal{E}$ occurs (meaning $\mathbf{C}$ is a $(1 \pm \gamma)$-approximate strong coreset for $C_{-1} \cup C_{-2}$). The second is by the definition of the $\ell_{p,2}$ norm (since after raising it to the p^{th} power, it decomposes across columns). The third equality is by the induction hypothesis, and the last equality is again because the p^{th} power of the $\ell_{p,2}$ norm decomposes across columns. This completes the proof of Lemma 5. □

Finally, using Lemma 5, we give a full analysis of our single-pass streaming algorithm, **Algorithm 2**.

Theorem 2 (A One-pass Streaming Algorithm for k-CSS$_p$)**.** *In the column-update streaming model, let $A \in \mathbb{R}^{d \times n}$ be the data matrix whose columns arrive one at each time in a data stream. Given $p \in [1, 2)$ and a desired rank $k \in \mathbb{N}$, Algorithm 2 outputs a subset of columns $A_I \in \mathbb{R}^{d \times k poly(\log(k))}$ in $\widetilde{O}(nnz(A)k + nk + k^3)$ time, such that with probability $1 - o(1)$,*

$$\min_V \|A_I V - A\|_p \le \widetilde{O}(k^{1/p-1/2}) \min_{L \subset [n], |L| = k} \|A_L V - A\|_p$$

Moreover, Algorithm 2 only needs to process all columns of A once and uses $\widetilde{O}(dk)$ space throughout the stream.

Proof. **Approximation Factor:** Note that n/r, the number of leaves, might not be a power of 2, and so we might get a list of coresets instead of a single one at the end of the stream. Consider the list C, of coresets and their corresponding level numbers, left at the end of the stream, before Algorithm 2 applies **Recursive Merge** after the data stream for the last time (to get a single output coreset). Denote these coresets by $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_t$, where $t = |C|$.

First, since strong coresets are subsets consisting of subsampled and reweighted columns of SA, we can let SAT denote the concatenation of the $\mathbf{C}_i$, where T is a sampling and reweighting matrix. In addition, note that for each $\mathbf{C}_i$, the subtree rooted at $\mathbf{C}_i$ has depth at most $\log(n/r)$, since all leaves represent contiguous blocks of r columns, and each coreset also has r columns. We bound $\min_V \|A_I V - A\|_p$ using Lemma 5 and these observations. In the following, let $L \subset [n]$, $|L| = k$ denote the subset of k columns of A that gives the minimum k-CSS$_p$ cost, i.e. the one minimizing $\min_V \|A_L V - A\|_p$. First note that with probability $1 - o(1)$,

$$
\begin{aligned}
\min_V \|A_I V - A\|_p &\le \|A_I V' - A\|_p && (\text{where } V' = \arg\min_V \|SA_I V - SA\|_{p,2}) \\
&\le \|SA_I V' - SA\|_p && \text{By } \textbf{Lemma } 2 \\
&= \widetilde{O}(k^{\frac{1}{p} - \frac{1}{2}}) \|SA_I V' - SA\|_{p,2} && \text{By } \textbf{Lemma } 1
\end{aligned}
$$

where I is the subset of column indices output by the bi-criteria $O(1)$-approximation algorithm for k-CSS$_{p,2}$ that we apply at the end of Algorithm 2. Let $(SAT)^*$ denote the best rank k approximation to SAT in the $\ell_{p,2}$-norm. By **Theorem** 1, with probability $1 - o(1)$,

$$
\begin{aligned}
\widetilde{O}(k^{\frac{1}{p} - \frac{1}{2}}) \|SA_I V' - SA\|_{p,2} &\le \widetilde{O}(k^{\frac{1}{p} - \frac{1}{2}}) \cdot O(1) \|(SAT)^* - SAT\|_{p,2} \\
&\le \widetilde{O}(k^{\frac{1}{p} - \frac{1}{2}}) \min_V \|SA_L V - SAT\|_{p,2}
\end{aligned}
$$

Now, recall that SAT is the concatenation of the coresets $\mathbf{C}_i$ — since the p^{th} power of the $\ell_{p,2}$ norm decomposes across columns,

$$\min_V \|SA_L V - SAT\|_{p,2}^p = \sum_{i=1}^t \min_V \|SA_L V - \mathbf{C}_i\|_{p,2}^p$$

Suppose that the subtree rooted at $\mathbf{C}_i$ has q_i nodes, and has depth l_i, and in addition, let M_i be the contiguous range of columns of SA which are represented by the leaves of the subtree rooted at $\mathbf{C}_i$. Then, by the second statement of Lemma 5, with probability at least $1 - \frac{q_i}{n^2}$,

$$\min_V \|SA_L V - \mathbf{C}_i\|_{p,2} = (1 \pm \gamma)^{l_i} \min_V \|SA_L V - M_i\|_{p,2}$$

Thus, by a union bound, this occurs simultaneously for *all* $i \in [t]$ with probability at least $1 - \sum_{i=1}^t \frac{q_i}{n^2} = 1 - \frac{1}{n}$ (since the subtrees rooted at the $\mathbf{C}_i$'s together contain all coresets ever created by the streaming algorithm). Thus, with probability at least $1 - \frac{1}{n}$,

$$\begin{aligned}
\min_V \|SA_L V - SAT\|_{p,2}^p &= \sum_{i=1}^t \min_V \|SA_L V - \mathbf{C}_i\|_{p,2}^p \\
&= \sum_{i=1}^t (1 \pm \gamma)^{pl_i} \min_V \|SA_L V - M_i\|_{p,2}^p \\
&= (1 \pm \gamma)^{p \log(n/r)} \min_V \|SA_L V - SA\|_{p,2}^p
\end{aligned} \tag{1}$$

where the last equality is because SA is the concatenation of the M_i, by their definition, and the subtree rooted at $\mathbf{C}_i$ has depth at most $\log(n/r)$. Taking p^{th} roots, with probability $1 - 1/n$,

$$\min_V \|SA_L V - SAT\|_{p,2} = (1 \pm \gamma)^{\log(n/r)} \min_V \|SA_L V - SA\|_{p,2}$$

Setting $\gamma = \frac{\varepsilon}{2\log(n/r)}$, we obtain the following with probability $1 - 1/n$,

$$\min_V \|SA_L V - SAT\|_{p,2} = (1 \pm \varepsilon) \min_V \|SA_L V - SA\|_{p,2}$$

Thus, by a union bound over all the events, with probability $1 - o(1)$,

$$\begin{aligned}
\widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_V \|SA_L V - SAT\|_{p,2} &\leq \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_V \|SA_L V - SA\|_{p,2} \\
&\leq \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_V \|SA_L V - SA\|_p && \text{By } \textbf{Lemma } 1 \\
&\leq \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \cdot \log^{1/p}(nd) \min_V \|A_L V - A\|_p && \text{By } \textbf{Lemma } 3
\end{aligned}$$

and we conclude that $\min_V \|A_I V - A\|_p \leq \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_V \|A_L V - A\|_p$ with probability $1 - o(1)$.

Space Complexity: Since the nodes are merged greedily during the data stream, and within the list C are in decreasing order according to their level, at most one node at each level l is in the list C at any time. Since the number of columns at each node in the binary tree is $\widetilde{O}(k)$ (i.e. the size of one coreset), the total space complexity is $\widetilde{O}(kd)$, suppressing logarithmic factors in n, d, k.

Running time: Since generating a single p-stable random variable takes $O(1)$ time, generating the dense p-stable sketching matrix S takes $O(dk \cdot \text{poly}(\log(nd)))$ time. Computing $SA_{*j}, \forall j \in [n]$ takes a total of $O(\text{nnz}(A) \cdot k\text{poly}(\log(nd)))$ time. By Lemma 4, merging two coresets, which are matrices of size $\widetilde{O}(k) \times \widetilde{O}(k)$, takes $\widetilde{O}(k^2)$ time. The merging operation is performed at most $O(n/k)$ times, so the total time it takes for merging is $\widetilde{O}(nk)$. By Theorem 1, the k-CSS$_{p,2}$ algorithm takes at most $k^3\text{poly}(\log(knd))$ time to find the final subset of columns. Since the number of selected columns is $k\text{poly}(\log k)$, it takes $k\text{poly}(\log k)$ time to map the indices and recover the original columns A_I. Therefore, the overall running time is $\widetilde{O}(\text{nnz}(A)k + nk + k^3)$, suppressing a low degree polynomial dependency on $\log(knd)$. □

D The Distributed Protocol and Full Analysis (Section 5)

We give our one-round distributed protocol for k-CSS$_p$ ($1 \le p < 2$) in **Algorithm** 4 and the full analysis below.

Algorithm 4 A one-round protocol for bi-criteria k-CSS$_p$ in the column partition model

```
**Initial State:**
Server $i$ holds matrix $A_i \in \mathbb{R}^{d\times n_i}$, $\forall i \in [s]$.
**Coordinator:**
Generate a dense $p$-stable sketching matrix $S \in \mathbb{R}^{k\ \mathrm{poly}(\log(nd))\times d}$.
Send $S$ to all servers.
**Server $i$:**
Compute $SA_i$.
Let the number of samples in the coreset be $t = O(k\mathrm{poly}(\log(nd))\log(1/\delta))$. Construct a coreset of $SA_i$ under the $\ell_{p,2}$ norm by applying a sampling matrix $D_i$ of size $n_i \times t$ and a diagonal reweighting matrix $W_i$ of size $t \times t$.
Let $T_i = D_iW_i$. Send $SA_iT_i$ along with $A_iD_i$ to the coordinator.
**Coordinator:**
Column-wise stack $SA_iT_i$ to obtain $SAT = [SA_1T_1, SA_2T_2, \ldots, SA_sT_s]$.
Apply $k$-CSS$_{p,2}$ on $SAT$ to obtain the indices $I$ of the subset of selected columns with size $O(k \cdot \mathrm{poly}(\log k))$.
Since $D_i$'s are sampling matrices, the coordinator can recover the original columns of $A$ by mapping indices $I$ to $A_iD_i$'s.
Denote the final selected subset of columns by $A_I$. Send $A_I$ to all servers.
**Server $i$:**
Solve $\min_{V_i} \|A_IV_i - A_i\|_p$ to obtain the right factor $V_i$. $A_I$ and $V$ will be factors of a rank-$k \cdot \mathrm{poly}(\log k)$ factorization of $A$, where $V$ is the (implicit) column-wise concatenation of the $V_i$.
```

Theorem 3 (A One-round Protocol for Distributed k-CSS$_p$)**.** *In the column partition model, let $A \in \mathbb{R}^{d\times n}$ be the data matrix whose columns are partitioned across s servers and suppose server i holds a subset of columns $A_i \in \mathbb{R}^{d\times n_i}$, where $n = \sum_{i\in[s]} n_i$. Then, given $p \in [1, 2)$ and a desired rank $k \in \mathbb{N}$, Algorithm 4 outputs a subset of columns $A_I \in \mathbb{R}^{d\times kpoly(log(k))}$ in $\widetilde{O}(nnz(A)k + kd + k^3)$ time, such that with probability $1 - o(1)$,*

$$\min_V \|A_IV - A\|_p \le \widetilde{O}(k^{1/p-1/2}) \min_{L\subset[n],|L|=k} \|A_LV - A\|_p$$

Moreover, Algorithm 4 uses one round of communication and $\widetilde{O}(sdk)$ words of communication.

Proof. **Approximation Factor:** In the following proof, let $L \subset [n], |L| = k$ denote the best possible subset of k columns of A that gives the minimum k-CSS$_p$ cost, i.e., the cost $\min_V \|A_LV - A\|_p$ achieves minimum. First, note that with probability $1 - o(1)$,

$$\begin{aligned}\min_V \|A_IV - A\|_p &\le \|A_IV' - A\|_p && V' := \arg\min_V \|SA_IV - SA\|_{p,2}\\ &\le \|SA_IV' - SA\|_p && \text{By } \mathbf{Lemma}\ 2\\ &= \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}})\|SA_IV' - SA\|_{p,2} && \text{By } \mathbf{Lemma}\ 1\end{aligned}$$

SA_I is the selected columns output from the bi-criteria $O(1)$-approximation k-CSS$_{p,2}$ algorithm. Let $(SAT)^*$ denote the best rank k approximation to SAT. By **Theorem** 1, with probability $1 - o(1)$,

$$\begin{aligned}\widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}})\|SA_IV' - SA\|_{p,2} &\le \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \cdot O(1)\|(SAT)^* - SAT\|_{p,2}\\ &\le \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_V \|SA_LV - SAT\|_{p,2}\end{aligned}$$

Note that $SAT = [SA_1T_1, \ldots, SA_sT_s]$ is a column-wise concatenation of all coresets of SA_i, $\forall i \in [s]$. By **Lemma** 4, and a union bound over the $i \in [s]$, with probability $1 - s\delta = 1 - o(1)$,

$$(\min_V \|SA_LV - SAT\|_{p,2}^p)^{1/p} = (\sum_{i=1}^s \min_{V_i} \|SA_LV_i - SA_iT_i\|_{p,2}^p)^{1/p}$$

$$
\begin{aligned}
&= (\sum_{i=1}^{s} (1 \pm \epsilon)^p \min_{V_i} \|SA_L V_i - SA_i\|_{p,2}^p)^{1/p} \\
&= (1 \pm \epsilon)(\sum_{i=1}^{s} \min_{V_i} \|SA_L V_i - SA_i\|_{p,2}^p)^{1/p} \\
&= (1 \pm \epsilon) \min_{V} \|SA_L V - SA\|_{p,2}
\end{aligned}
$$

Hence, by a union bound over all the events, with probability $1 - o(1)$,

$$
\begin{aligned}
\widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_{V} \|SA_L V - SAT\|_{p,2} &\leq \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_{V} \|SA_L V - SA\|_{p,2} \\
&\leq \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_{V} \|SA_L V - SA\|_{p} && \text{By } \textbf{Lemma } 1 \\
&\leq \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \cdot \log^{1/p}(nd) \min_{V} \|A_L V - A\|_{p} && \text{By } \textbf{Lemma } 3
\end{aligned}
$$

Thus, $\min_V \|A_I V - A\|_p \leq \widetilde{O}(k^{\frac{1}{p}-\frac{1}{2}}) \min_V \|A_L V - A\|_p$ with probability $1 - o(1)$.

Communication Cost: Sharing the dense p-stable sketching matrix S with all servers costs $O(sdk \cdot \text{poly}(\log(nd)))$ communication (this can be removed with a shared random seed). Sending all coresets SA_iT_i ($\forall i \in [s]$) and the corresponding columns A_iD_i to the coordinator costs $\widetilde{O}(sdk)$ communication, since each coreset contains only $\widetilde{O}(k)$ columns (note that since we compute s coresets, each coreset computation should have a failure probability of $\frac{1}{\text{poly}(s)}$ to allow us to union bound — this only increases the communication cost by a $\log(s)$ factor, however). Finally, the coordinator needs $\widetilde{O}(sdk)$ words of communication to send the $\widetilde{O}(k)$ selected columns to each server. Therefore, the overall communication cost is $\widetilde{O}(sdk)$, suppressing a logarithmic factor in n, d.

Running time: Since generating a single p-stable random variable takes $O(1)$ time, generating the dense p-stable sketching matrix S takes $O(dk \cdot \text{poly}(\log(nd)))$ time. Computing all SA_i's takes $O(\text{nnz}(A)k \cdot \text{poly}(\log(nd)))$ time. By Lemma 4, computing all coresets for $SA_iT_i, \forall i \in [s]$ takes time $\widetilde{O}(kd)$. By Theorem 1, the k-CSS$_{p,2}$ algorithm takes time $(\text{nnz}(SAT) + k^2\text{poly}(\log nd)) \cdot k\text{poly}(\log k) \leq k^3\text{poly}(\log(knd))$ to find the set of selected columns. Since the number of selected columns is $O(k\text{poly}(\log k))$, it then takes the protocol $O(k\text{poly}(\log k))$ time to map the indices and recover the original columns A_I. Therefore, the overall running time is $\widetilde{O}((\text{nnz}(A)k + kd + k^3)$, suppressing a low degree polynomial dependency on $\log(knd)$. After the servers receive A_I, it is possible to solve $\min_{V_i} \|A_I V_i - A_i\|_p$ in $\widetilde{O}(\text{nnz}(A_I)) + \text{poly}(d \log n)$ time , $\forall i \in [s]$ due to [14, 15]. □

E A High Communication Cost Protocol for k-CSS$_p$ ($p \geq 1$) (Section 1)

We describe in detail the naive protocol for distributed k-CSS$_p$ mentioned in Section 1, which works for all $p \geq 1$, in the column partition model, and which achieves an $O(k^2)$-approximation to the best rank-k approximation, using $O(1)$ rounds and polynomial time but requiring a communication cost that is linear in $n + d$. The inputs are a column-wise partitioned data matrix $A \in \mathbb{R}^{d \times n}$ distributed across s servers and a rank parameter $k \in \mathbb{N}$. Each server i holds part of the data matrix $A_i \in \mathbb{R}^{d \times n_i}$, $\forall i \in [s]$, and such that $\sum_{i=1}^{s} n_i = n$.

We use a single machine, polynomial time bi-criteria k-CSS$_p$ algorithm as a subroutine of the protocol, e.g., Algorithm 3 in [16], which selects a subset of $\widetilde{O}(k)$ columns A_T of the data matrix $A \in \mathbb{R}^{d \times n}$ in polynomial time, for which $\min_X \|A_T X - A\|_p \leq O(k) \min_{\text{rank-k } A_k} \|A - A_k\|_p$, $\forall p \geq 1$.

Algorithm 5 A protocol for k-CSS$_p$ ($p \geq 1$)

```
Initial State: Server $i$ holds matrix $A_i \in \mathbb{R}^{d \times n_i}$, $\forall i \in [s]$.
Server $i$:
Apply polynomial time bi-criteria $k$-CSS$_p$ on $A_i$ to obtain a subset $B_i$ of columns as the left factor.
Solve for the right factor $V_i = \arg\min_{V_i} \|U_i V_i - A_i\|_p$. Send $U_i$ and $V_i$ to the coordinator.
Coordinator:
Column-wise concatenate the $U_i V_i$ to obtain $UV = [U_1 V_1, \ldots, U_s V_s]$. Apply a polynomial time bi-criteria $k$-CSS$_p$ algorithm on $UV$ to obtain a subset $C$ of columns. Send $C$ to each server.
Server $i$:
Solve $\min_{X_i} \|C X_i - A_i\|_p$ to obtain the right factor.
```

Approximation Factor. Let UV denote the column-wise concatenation of the $U_i V_i$. Let $X^* = \arg\min_X \|CX - A\|_p$. Then,

$$\begin{aligned}
\|CX^* - AS\|_p &\leq \|CX^* - UV\|_p + \|UV - A\|_p && \text{By the triangle inequality} \\
&\leq O(k) \min_{\text{rank-k } (UV)_k} \|UV - (UV)_k\|_p + \|UV - A\|_p && \text{By the } O(k)\text{-approximation of } k\text{-CSS}_p \\
&\leq O(k)\|UV - A\|_p \\
&= O(k)\Big(\sum_{i=1}^{s} \|U_i V_i - A_i\|_p\Big) \\
&\leq O(k)\Big(\sum_{i=1}^{s} O(k) \min_{\text{rank-k } A_i^*} \|A_i - A_i^*\|_p\Big) && \text{By the } O(k)\text{-approximation of } k\text{-CSS}_p \\
&\leq O(k^2) \sum_{i=1}^{s} \|A_i - (A^*)_i\|_p && A^* = \arg \min_{\text{rank-k } A^*} \|A - A^*\|_p \\
&= O(k^2)\|A - A^*\|_p
\end{aligned}$$

Communication Cost. Since $U_i \in \mathbb{R}^{d \times \widetilde{O}(k)}$ and $V_i \in \mathbb{R}^{\widetilde{O}(k) \times n_i}$, sending U_i and V_i costs $\widetilde{O}(skn)$. Since $C \in \mathbb{R}^{d \times \widetilde{O}(k)}$, sending C from the coordinator to all servers costs $\widetilde{O}(sdk)$. Thus the overall communication cost is $\widetilde{O}(s(n + d)k)$.

Running time. According to [16], applying the k-CSS$_p$ algorithm and solving ℓ_p regression can both be done in polynomial time. Thus the overall running time of the protocol is polynomial.

Problems with this protocol. Although this protocol works for all $p \geq 1$, a communication cost that linearly depends on the large dimension n is too high, and furthermore, the output C is not a subset of columns of A, because the protocol applies k-CSS$_p$ on a concatenation of both the left factor U_i and the right factor V_i. U_i is a subset of columns of A_i but V_i is not necessarily a sampling matrix. One might wonder whether it is possible that each server only sends U_i and the coordinator then runs k-CSS$_p$ on a concatenation of the U_i. This will not necessarily give a good approximation to $\min_{\text{rank-k } A_k} \|A - A_k\|_p$ because the columns not selected in the U_i locally on each server might become globally important. Finally, although it is possible to improve the approximation factor to $\widetilde{O}(k)$ by making use of an $\widetilde{O}(\sqrt{k})$-approximation algorithm for ℓ_p-low rank approximation that also selects a subset of columns [17], this protocol would still suffer from all of the aforementioned problems.

F Greedy k-CSS$_{p,2}$ and Full Analysis (Section 6)

Algorithm 6 Greedy k-CSS$_{p,2}$.

```
Input: The data matrix $A \in \mathbb{R}^{d\times n}$. A desired rank $k \in \mathbb{N}$ and $p \in [1,2)$. The number of columns
to be selected $r \le n$. Failure probability $\delta \in (0,1)$.
Output: A subset of $r$ columns $A_T$.
Indices of selected columns $T \leftarrow \{\}$.
for $i = 1$ to $r$ do
    $C \leftarrow$ Sample $\frac{n}{k}\log(\frac{1}{\delta})$ indices from $\{1,2,\dots,n\} \setminus T$ uniformly at random.
    Column index $j^* \leftarrow \arg\min_{j\in C}(\min_V \|A_{T\cup j}V - A\|_{p,2})$
    $T \leftarrow T \cup j^*$.
end for
Map indices $T$ to get the selected columns $A_T$.
```

We propose a greedy algorithm for selecting columns in k-CSS$_{p,2}$ (**Algorithm** 6) for $p \in [1,2)$. We give a detailed analysis on the first additive approximation compared to the error of the optimal column subset for Greedy k-CSS$_{p,2}$. Our analysis is inspired by the analysis of the Frobenius norm Greedy k-CSS$_2$ algorithm in [18].

Notice that during each iteration, the algorithm needs to evaluate the error $\min_V \|A_{T\cup j}V - A\|_{p,2}$ to greedily pick the next column j. A standard greedy algorithm which considers all unselected columns in $[n]\setminus T$ would need $O(nr)$ evaluations of the regression error $\min_V \|A_{T\cup j}V - A\|_{p,2}$, which is too expensive. To improve the running time, we adopt the *Lazier-than-lazy* framework for greedy algorithms originally introduced in [19] and used by [18] in greedy k-CSS$_2$. Instead of considering all unselected columns at each iteration, we first randomly sample $\frac{n}{k}\log(\frac{1}{\delta})$ candidate columns from $[n]\setminus T$ and greedily pick the next column only among those candidates. This reduces the number of evaluations of $\min_V \|A_{T\cup j}V - A\|_{p,2}$ to $O(n\log(\frac{1}{\delta}))$.

To aid the analysis, we first define a utility function that quantifies how well the selected columns approximate the original matrix in **Notation** below as in [18]. We show in **Lemma** 7.2 an improvement of the utility function with one additional column when projecting a single vector, based on **Lemma** 7.1 from [18]. We then show an improvement of the utility function when projecting a matrix in **Lemma** 7.3, by applying **Lemma** 7.2 and Jensen's Inequality, following the analysis in [18]. With **Lemma** 7.3, we show a large expected improvement in the utility function by choosing a column from a subsampled candidates, based on **Lemma 6** from [18]. Finally, we conclude by giving the convergence rate and the running time for *Lazier-than-lazy* based Greedy k-CSS$_{p,2}$ in **Theorem** 4.

Notation Consider the input matrix $A \in \mathbb{R}^{d\times n}$ ($n \gg d$). Let B be the matrix of normalized columns of A, where the j-th column of B is $B_{*j} = A_{*j}/\|A_{*j}\|_2$. Let π_T be the projection matrix onto the column span of A_T or equivalently B_T. Let $\sigma_{\min}(M)$ denote the minimum singular value of some matrix M.

To aid our analysis, we define a *utility function* Φ as follows, inspired by [18]. For a subset $T \subset [n]$ and a matrix $M \in \mathbb{R}^{d\times t}$ (or a vector $M \in \mathbb{R}^d$),

$$\Phi_M(T) = \|M\|_{p,2}^p - \|M - \pi_T M\|_{p,2}^p = \sum_{i=1}^{t}\left(\|M_{*i}\|_2^p - \|M_{*i} - \pi_T M_{*i}\|_2^p\right) = \sum_{i=1}^{t}\Phi_{M_{*i}}(T)$$

Observe that as the number of columns selected and added to T increases, we get a more accurate estimation of M and thus the approximation error $\|M - \pi_T M\|_{p,2}$ decreases, which results in an increase in the utility function $\Phi_M(T)$.

Lemma 7.1. *Let $S, T \subset [n]$ be two sets of column indices, with $S = \{i_1,\dots,i_k\}$ and $\|\pi_S u\|_2 \ge \|\pi_T u\|_2$ for some vector $u \in \mathbb{R}^d$. Then,*

$$\sum_{j=1}^{k}\left(\|\pi_{T'_j}u\|_2^2 - \|\pi_T u\|_2^2\right) \ge \sigma_{min}(B_S)^2\frac{(\|\pi_S u\|_2^2 - \|\pi_T u\|_2^2)^2}{4\|\pi_S u\|_2^2}$$

where $T'_j = T \cup \{i_j\}$ for all $j \in [k]$.

Proof. **Lemma 2** from [18], except that we replace the condition for S and T, i.e., $\Phi_u(S) \geq \Phi_u(T)$ in [18] with $\|\pi_S u\|_2 \geq \|\pi_T u\|_2$. The two conditions are equivalent, since

$$\begin{aligned}
&\Phi_u(S) \geq \Phi_u(T) \\
\Leftrightarrow\ &\|u\|_2^p - \|u - \pi_S u\|_2^p \geq \|u\|_2^p - \|u - \pi_T u\|_2^p \\
\Leftrightarrow\ &\|u - \pi_S u\|_2 \leq \|u - \pi_T u\|_2 \\
\Leftrightarrow\ &\|u\|_2^2 - \|\pi_S u\|_2^2 \leq \|u\|_2^2 - \|\pi_T u\|_2^2 \\
\Leftrightarrow\ &\|\pi_S u\|_2 \geq \|\pi_T u\|_2
\end{aligned}$$

□

Lemma 7.2. *(Utility Improvement by Projecting a Single Vector) Give $p \in [1, 2)$. Let $S, T \subset [n]$ be two sets of column indices, with $\Phi_u(S) \geq \Phi_u(T)$ for some vector $u \in \mathbb{R}^d$. Let $k = |S|$, and for $i \in S$, let $T_i' = T \cup \{i\}$. Then,*

$$\sum_{i \in S} \Big(\Phi_u(T_i') - \Phi_u(T)\Big) \geq \frac{p\sigma_{min}(B_S)^2}{16} \cdot \frac{(\Phi_u(S) - \Phi_u(T))^{2/p+1}}{\Phi_u(S)^{2/p}}$$

Proof. To aid the analysis, we define the decreasing function $g : (-\infty, \|u\|_2^2] \to \mathbb{R}$ by

$$g(x) = (\|u\|_2^2 - x)^{p/2}$$

and the derivative of g is

$$|g'(x)| = \frac{p}{2}(\|u\|_2^2 - x)^{p/2-1}$$

which is an increasing function for $p < 2$. Then,

$$\begin{aligned}
\sum_{i=1}^{k} \Big(\Phi_u(T_i') - \Phi_u(T)\Big) &= \sum_{i=1}^{k} \Big(\|u - \pi_T u\|_2^p - \|u - \pi_{T_i'} u\|_2^p\Big) && \text{By definition of } \Phi \\
&= \sum_{i=1}^{k} \Big((\|u\|_2^2 - \|\pi_T u\|_2^2)^{p/2} - (\|u\|_2^2 - \|\pi_{T_i'} u\|_2^2)^{p/2}\Big) && \text{By Pythagorean Theorem} \\
&= \sum_{i=1}^{k} \Big(g(\|\pi_T u\|_2^2) - g(\|\pi_{T_i'} u\|_2^2)\Big) && \text{By definition of } g \\
&\geq \sum_{i=1}^{k} |g'(\|\pi_T u\|_2^2)| \Big(\|\pi_{T_i'} u\|_2^2 - \|\pi_T u\|_2^2\Big) && \text{Mean Value Theorem and} \\
& && \|\pi_T u\|_2 \leq \|\pi_{T_i'} u\|_2 \\
&= |g'(\|\pi_T u\|_2^2)| \sum_{i=1}^{k} \Big(\|\pi_{T_i'} u\|_2^2 - \|\pi_T u\|_2^2\Big) \\
&= \frac{p}{2}\Big(\|u\|_2^2 - \|\pi_T u\|_2^2\Big)^{p/2-1} \sum_{i=1}^{k} \Big(\|\pi_{T_i'} u\|_2^2 - \|\pi_T u\|_2^2\Big) \\
&= \frac{p}{2}\|u - \pi_T u\|_2^{p-2} \sum_{i=1}^{k} \Big(\|\pi_{T_i'} u\|_2^2 - \|\pi_T u\|_2^2\Big) \\
&\geq \frac{p}{2}\|u - \pi_T u\|_2^{p-2} \cdot \sigma_{min}(B_S)^2 \frac{(\|\pi_S u\|_2^2 - \|\pi_T u\|_2^2)^2}{4\|\pi_S u\|_2^2} && \text{Lemma 7.1} \\
&= \frac{p\sigma_{min}(B_S)^2}{2} \cdot \frac{(\|u - \pi_T u\|_2^2 - \|u - \pi_S u\|_2^2)^2}{4\|\pi_S u\|_2^2 \|u - \pi_T u\|_2^{2-p}}
\end{aligned}$$

Now we can lower bound

$$\begin{aligned}
\|u - \pi_T u\|_2^2 - \|u - \pi_S u\|_2^2 &= \|u - \pi_T u\|_2^{2-p}\|u - \pi_T u\|_2^p - \|u - \pi_S u\|_2^{2-p}\|u - \pi_S u\|_2^p \\
&\geq \|u - \pi_T u\|_2^{2-p}\Big(\|u - \pi_T u\|_2^p - \|u - \pi_S u\|_2^p\Big) && \text{since } \|u - \pi_S u\|_2 \leq \|u - \pi_T u\|_2 \\
&= \|u - \pi_T u\|_2^{2-p}\Big(\Phi_u(S) - \Phi_u(T)\Big)
\end{aligned}$$

Thus,

$$\begin{aligned}
\sum_{i=1}^{k}\Big(\Phi_u(T_i') - \Phi_u(T)\Big) &\geq \frac{p\sigma_{min}(B_S)^2}{2} \cdot \frac{(\|u-\pi_T u\|_2^2 - \|u-\pi_S u\|_2^2)^2}{4\|\pi_S u\|_2^2 \|u-\pi_T u\|_2^{2-p}} \\
&\geq \frac{p\sigma_{min}(B_S)^2}{2} \cdot \frac{\Big(\Phi_u(S)-\Phi_u(T)\Big)^2 \cdot \|u-\pi_T u\|_2^{2(2-p)}}{4\|\pi_S u\|_2^2 \|u-\pi_T u\|_2^{2-p}} \\
&= \frac{p\sigma_{min}(B_S)^2}{2} \cdot \frac{\Big(\Phi_u(S)-\Phi_u(T)\Big)^2 \cdot \|u-\pi_T u\|_2^{2-p}}{4\|\pi_S u\|_2^2}
\end{aligned}$$

Now to finish the proof, let us lower bound $\frac{\|u-\pi_T u\|_2^{2-p}}{\|\pi_S u\|_2^2}$. First, observe that $\|u\|_2^p\|u-\pi_S u\|_2^{2-p} - \|u\|_2^{2-p}\|u-\pi_S u\|_2^p \geq 0$. To see why, observe that the following equivalences hold:

$$\begin{aligned}
&\|u\|_2^p\|u-\pi_S u\|_2^{2-p} \geq \|u\|_2^{2-p}\|u-\pi_S u\|_2^p \\
\Leftrightarrow\ &\|u\|_2^p/\|u\|_2^{2-p} \geq \|u-\pi_S u\|_2^p/\|u-\pi_S u\|_2^{2-p} \\
\Leftrightarrow\ &\|u\|_2^{2p-2} \geq \|u-\pi_S u\|_2^{2p-2}
\end{aligned}$$

and the last statement is true, since $\|u\|_2 \geq \|u-\pi_S u\|_2$ and $f(x) = x^{2p-2}$ is a monotone function. Thus,

$$\begin{aligned}
\frac{\|u-\pi_T u\|_2^{2-p}}{\|\pi_S u\|_2^2} &= \frac{\|u-\pi_T u\|_2^{2-p}}{\|u\|_2^2 - \|u-\pi_S u\|_2^2} \\
&\geq \frac{\|u-\pi_T u\|_2^{2-p}}{\|u\|_2^2 - \|u-\pi_S u\|_2^2 + \|u\|_2^p\|u-\pi_S u\|_2^{2-p} - \|u\|_2^{2-p}\|u-\pi_S u\|_2^p} \\
&= \frac{\|u-\pi_T u\|_2^{2-p}}{(\|u\|_2^p - \|u-\pi_S u\|_2^p)(\|u\|_2^{2-p} + \|u-\pi_S u\|_2^{2-p})} \\
&= \frac{1}{\Phi_u(S)} \cdot \frac{\|u-\pi_T u\|_2^{2-p}}{\|u\|_2^{2-p} + \|u-\pi_S u\|_2^{2-p}} \\
&\geq \frac{1}{2\Phi_u(S)} \cdot \frac{\|u-\pi_T u\|_2^{2-p}}{\|u\|_2^{2-p}} && \text{Since } \|u\|_2^{2-p} \geq \|u-\pi_S u\|_2^{2-p} \\
&= \frac{1}{2\Phi_u(S)} \cdot \Big(\frac{\|u-\pi_T u\|_2^p}{\|u\|_2^p}\Big)^{2/p-1} \\
&= \frac{1}{2\Phi_u(S)} \cdot \Big(\frac{\|u\|_2^p - \Phi_u(T)}{\|u\|_2^p}\Big)^{2/p-1} && \text{By definition of } \Phi \\
&\geq \frac{1}{2\Phi_u(S)} \cdot \Big(1 - \frac{\Phi_u(T)}{\Phi_u(S)}\Big)^{2/p-1} && \text{Since } \Phi_u(S) \leq \|u\|_2^p \\
&= \frac{(\Phi_u(S)-\Phi_u(T))^{2/p-1}}{2\Phi_u(S)^{2/p}}
\end{aligned}$$

Combining all the above inequalities gives

$$\begin{aligned}
\sum_{i=1}^{k}\Big(\Phi_u(T_i') - \Phi_u(T)\Big) &\geq \frac{p\sigma_{min}(B_S)^2}{2} \cdot \frac{\Big(\Phi_u(S)-\Phi_u(T)\Big)^2 \cdot \|u-\pi_T u\|_2^{2-p}}{4\|\pi_S u\|_2^2} \\
&\geq \frac{p\sigma_{min}(B_S)^2}{8} \cdot \Big(\Phi_u(S)-\Phi_u(T)\Big)^2 \cdot \frac{(\Phi_u(S)-\Phi_u(T))^{2/p-1}}{2\Phi_u(S)^{2/p}} \\
&= \frac{p\sigma_{min}(B_S)^2}{16} \cdot \frac{(\Phi_u(S)-\Phi_u(T))^{2/p+1}}{\Phi_u(S)^{2/p}}
\end{aligned}$$

This completes the proof. □

Lemma 7.3. *(Utility Improvement by Projecting a Matrix) Given $p \in [1,2)$. Let $A \in \mathbb{R}^{d\times n}$, and $T, S \subset [n]$ be two sets of column indices, with $\Phi_A(S) \geq \Phi_A(T)$. Furthermore, let $k = |S|$. Then,* *there exists a column index $i \in S$ such that*

$$\Phi_A(T \cup \{i\}) - \Phi_A(T) \geq p\sigma_{min}(B_S)^2 \frac{(\Phi_A(S) - \Phi_A(T))^{2/p+1}}{16k\Phi_A(S)^{2/p}}$$

Proof. The proof mostly follows the proof of **Lemma 1** in [18]. We combine Lemma 7.2 with Jensen's inequality to conclude an improvement of the utility function with one additional column when projecting a matrix instead of a single column.

For $j \in [n]$, we define $\delta_j = \min(1, \frac{\Phi_{A_{*j}}(T)}{\Phi_{A_{*j}}(S)})$. Note that δ_j is 1 if the j-th column A_{*j} has a larger projection onto B_T than B_S, and $\frac{\Phi_{A_{*j}}(T)}{\Phi_{A_{*j}}(S)}$ otherwise. Note that $f(x) = x^{2/p+1}$ is convex on $x \in [0, \infty)$, $\forall 1 \leq p < 2$, based on which we will apply Jensen's inequality.

Let $k = |S|$. For $i \in [k]$, let $T'_i = T \cup \{i\}$.

$$\begin{aligned}
\frac{1}{p\sigma_{min}(B_S)^2} \sum_{i=1}^{k} \Big(\Phi_A(T'_i) - \Phi_A(T)\Big) &= \frac{1}{p\sigma_{min}(B_S)^2} \sum_{j=1}^{n} \sum_{i=1}^{k} \Big(\Phi_{A_{*j}}(T'_i) - \Phi_{A_{*j}}(T)\Big) && \text{By definition of } \Phi \\
&\geq \sum_{j=1}^{n} \frac{(1-\delta_j)^{2/p+1}}{16} \cdot \Phi_{A_{*j}}(S) && \text{By \textbf{Lemma} 7.2} \\
&= \frac{\Phi_A(S)}{16} \sum_{j=1}^{n} (1-\delta_j)^{2/p+1} \cdot \frac{\Phi_{A_{*j}}(S)}{\sum_{i=1}^{n} \Phi_{A_{*i}}(S)} && \text{Note } \Phi_A(S) = \sum_{i=1}^{n} \Phi_{A_{*i}}(S) \\
&\geq \frac{\Phi_A(S)}{16} \Big(\sum_{j=1}^{n} (1-\delta_j) \cdot \frac{\Phi_{A_{*j}}(S)}{\sum_{i=1}^{n} \Phi_{A_{*i}}(S)}\Big)^{2/p+1} && \text{By Jensen's Inequality} \\
&= \frac{1}{16\Phi_A(S)^{2/p}} \Big(\sum_{j=1}^{n} (1-\delta_j) \cdot \Phi_{A_{*j}}(S)\Big)^{2/p+1} \\
&\geq \frac{1}{16\Phi_A(S)^{2/p}} \Big(\sum_{j=1}^{n} (\Phi_{A_{*j}}(S) - \Phi_{A_{*j}}(T))\Big)^{2/p+1} && \text{Since } 1 - \delta_j \geq 1 - \frac{\Phi_{A_{*j}}(T)}{\Phi_{A_{*j}}(S)} \\
& && \Rightarrow (1-\delta_j) \cdot \Phi_{A_{*j}}(S) \\
& && \geq \Phi_{A_{*j}}(S) - \Phi_{A_{*j}}(T) \\
&= \frac{(\Phi_A(S) - \Phi_A(T))^{2/p+1}}{16\Phi_A(S)^{2/p}}
\end{aligned}$$

Hence,

$$\sum_{i=1}^{k} \Big(\Phi_A(T'_i) - \Phi_A(T)\Big) \geq p\sigma_{min}(B_S)^2 \frac{(\Phi_A(S) - \Phi_A(T))^{2/p+1}}{16\Phi_A(S)^{2/p}}$$

This implies there is at least one column of B_S, with index $i \in S$, such that when i is added to T, the utility function $\Phi_A(T)$ increases by at least $\frac{1}{k} \cdot p\sigma_{min}(B_S)^2 \frac{(\Phi_A(S) - \Phi_A(T))^{2/p+1}}{16\Phi_A(S)^{2/p}}$. □

Lemma 7.4 (Expected Increase in Utility). *Given $p \in [1, 2)$. Let $A \in \mathbb{R}^{d \times n}$, and let $T, S \subset [n]$ be two sets of column indices, with $k := |S|$ and $\Phi_A(S) \geq \Phi_A(T)$. Let $\overline{T}$ be a set of $\frac{n \log(1/\delta)}{k}$ column indices of A, chosen uniformly at random from $[n] \setminus T$. Then,*

$$\mathbb{E}[\max_{i \in \overline{T}} \Phi_A(T \cup \{i\})] - \Phi_A(T) \geq (1-\delta) \cdot p\sigma_{min}(B_S)^2 \cdot \frac{(\Phi_A(S) - \Phi_A(T))^{2/p+1}}{16k\Phi_A(S)^{2/p}}$$

Proof. The proof is nearly identical to the proof of **Lemma 6** of [18] — we include the full proof for completeness. The first step in the proof is showing that $\overline{T} \cap (S \setminus T)$ is nonempty with high probability. Then, by conditioning on $\overline{T} \cap (S \setminus T)$ being nonempty, we can show that the expected increase in utility is large. For the purpose of this analysis, we assume that the columns of $\overline{T}$ are sampled independently with replacement. At the end of the proof, we discuss sampling the columns of $\overline{T}$ without replacement.

First, observe that

$$\begin{aligned}
\Pr[\overline{T} \cap (S \setminus T) = \varnothing] &= \prod_{t=1}^{O\left(\frac{n\log(1/\delta)}{k}\right)} \Big(1 - \frac{|S \setminus T|}{n - |T|}\Big) \\
&= \Big(1 - \frac{|S \setminus T|}{n - |T|}\Big)^{O\left(\frac{n\log(1/\delta)}{k}\right)} \\
&\le e^{-\frac{|S\setminus T|}{n-|T|}\cdot \frac{n\log(1/\delta)}{k}} && \text{By } 1 - x \le e^{-x} \\
&\le e^{-\frac{|S\setminus T|\log(1/\delta)}{k}} && \text{Because } n - |T| < n
\end{aligned}$$

meaning that

$$\begin{aligned}
\Pr[\overline{T} \cap (S \setminus T) \neq \varnothing] &\ge 1 - e^{-\frac{|S\setminus T|\log(1/\delta)}{k}} \\
&= 1 - \delta^{\frac{|S\setminus T|}{k}} \\
&\ge (1-\delta)\frac{|S \setminus T|}{k} && \text{Since } |S \setminus T| \le k \text{, and } 1 - \delta^x \ge (1-\delta)x \text{ for } x, \delta \in [0,1]
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}[\max_{i \in \overline{T}} \Phi_A(T \cup \{i\}) - \Phi_A(T)] \\
&\ge \Pr[\overline{T} \cap (S \setminus T) \neq \varnothing] \cdot \mathbb{E}\Big[\max_{i\in\overline{T}} \Phi_A(T \cup \{i\}) - \Phi_A(T) \Big| \overline{T} \cap (S \setminus T) \neq \varnothing\Big] \\
&\ge (1-\delta)\frac{|S \setminus T|}{k} \cdot \mathbb{E}\Big[\max_{i\in\overline{T}} \Phi_A(T \cup \{i\}) - \Phi_A(T) \Big| \overline{T} \cap (S \setminus T) \neq \varnothing\Big] \\
&\ge (1-\delta)\frac{|S \setminus T|}{k} \cdot \mathbb{E}\Big[\max_{i\in\overline{T}} \Phi_A(T \cup \{i\}) - \Phi_A(T) \Big| |\overline{T} \cap (S \setminus T)| = 1\Big] \\
&\quad \text{(Since it is always better for } \overline{T} \cap (S \setminus T) \text{ to be larger)} \\
&= (1-\delta)\frac{|S \setminus T|}{k} \cdot \frac{\sum_{i \in S\setminus T}(\Phi_A(T \cup \{i\}) - \Phi_A(T))}{|S \setminus T|} \\
&\quad \text{(Since the single element of } \overline{T} \cap (S \setminus T) \text{ is uniformly random in } (S \setminus T)) \\
&= (1-\delta) \cdot \frac{\sum_{i \in S}(\Phi_A(T \cup \{i\}) - \Phi_A(T))}{|S|} \\
&\quad \text{(Since } \Phi_A(T \cup \{i\}) = \Phi_A(T) \text{ for } i \in T) \\
&\ge (1-\delta) \cdot \frac{1}{k} \cdot p\sigma_{min}(B_S)^2 \frac{(\Phi_A(S) - \Phi_A(T))^{2/p+1}}{16\Phi_A(S)^{2/p}} \\
&\quad \text{(By } \textbf{Lemma } 7.3.)
\end{aligned}$$

This proves the lemma in the case where the columns are sampled with replacement. Now, we discuss what happens when sampling without replacement. Note that the expected increase in utility can only be higher if the columns of $\overline{T}$ are sampled without replacement. Intuitively, this is because if $\overline{T}$ has some repeated columns, then it is always better to replace those repeated columns with other columns of A. Thus, for each instance of $\overline{T}$ where some columns are sampled multiple times, we can "move" all of the probability mass from this instance of $\overline{T}$ to other sets $\overline{T}' \subset [n] \setminus T$, which contain $\overline{T}$ but do not have repeated elements. This leads to the uniform distribution on subsets of $[n] \setminus T$ with no repeated elements, i.e., the distribution that results from sampling without replacement. □

Using this lemma, we analyze the convergence rate and the running time of **Algorithm** 6:

Theorem 4 (Greedy k-CSS$_{1,2}$)**.** *Let $p \in [1, 2)$. Let $A \in \mathbb{R}^{d\times n}$ be the data matrix and $k \in \mathbb{N}$ be the desired rank. Let A_L be the best possible subset of k columns, i.e., $A_L = \arg\min_{A_L} \min_V \|A_L V - A\|_{p,2}$. Let σ be the minimum non-zero singular value of the matrix B of normalized columns of A_L, (i.e., the j-th column of B is $B_{*j} = (A_L)_{*j}/\|(A_L)_{*j}\|_2$). Let $T \subset [n]$ be the subset of output column indices selected by **Algorithm** 6, for $\epsilon, \delta \in (0,1)$, for $|T| = \Omega(\frac{k}{p\sigma^2\epsilon^2})$, with probability $1-\delta$,*

$$\mathbb{E}[\min_V \|A_T V - A\|_{p,2}] \le \min_V \|A_L V - A\|_{p,2} + \epsilon\|A\|_{p,2}$$

The overall running time is $O(\frac{n}{p\sigma^2\epsilon^2}\log(\frac{1}{\delta}) \cdot (\frac{dk^2}{p^2\sigma^4\epsilon^4} + \frac{ndk}{p\sigma^2\epsilon^2}))$.

Proof. **Convergence Rate.** The proof uses the same strategy as that of Theorem 5 of [18], with minor modifications. Let $S := B_L$ be the best subset of columns of B. Let T_t be the subset of columns of B selected by **Algorithm** 6 after t iterations (in particular, $T_0 = \varnothing$). In addition, let $F = \Phi_A(S) = \Phi_A(S) - \Phi_A(T_0)$ be the distance from the current value of the utility function to the best achievable value. Let Δ_t denote a small amount at time t that is used to quantify our progress as how good the currently selected subset of columns approximates the matrix. When no column is selected, $\Delta_0 = F$. Let $\Delta_{i+1} = \frac{\Delta_i}{2}$. Now, fix a time t such that for some i, $\Delta_i \geq \Phi_A(S) - \Phi_A(T_t) \geq \Delta_{i+1} = \frac{\Delta_i}{2}$. Then, we bound the number of additional iterations t' needed so that

$$\mathbb{E}[\Phi_A(S) - \Phi_A(T_{t+t'}) \mid T_t] \leq \Delta_{i+1}$$

For convenience, for each $k \geq 0$, define $E_k := E[\Phi_A(T_{t+k}) \mid T_t]$. Then, our goal is to find t' such that

$$\Phi_A(S) - E_{t'} \leq \Delta_{i+1}$$

However, observe that from Lemma 7.4 above, we obtain

$$\begin{aligned}
E_{k+1} - E_k &= \mathbb{E}\Big[\Phi_A(T_{t+k+1}) - \Phi_A(T_{t+k})\Big|T_t\Big] \\
&= \mathbb{E}\Big[\mathbb{E}[\Phi_A(T_{t+k+1}) - \Phi_A(T_{t+k})|T_{t+k}]\Big|T_t\Big] && \text{By } \mathbb{E}[\mathbb{E}[X|Y]] = \mathbb{E}[X] \\
&\geq \mathbb{E}\Big[(1-\delta)\cdot p\sigma_{min}(B_S)^2 \cdot \frac{(\Phi_A(S) - \Phi_A(T_{t+k}))^{2/p+1}}{16k\Phi_A(S)^{2/p}}\Big|T_t\Big] && \text{By Lemma 7.4} \\
&= \frac{(1-\delta)\cdot p\sigma_{min}(B_S)^2}{16k\Phi_A(S)^{2/p}} \cdot \mathbb{E}\big[(\Phi_A(S) - \Phi_A(T_{t+k}))^{2/p+1}|T_t\big] \\
&\geq \frac{(1-\delta)\cdot p\sigma_{min}(B_S)^2}{16k\Phi_A(S)^{2/p}} \cdot \Big(\mathbb{E}[\Phi_A(S) - \Phi_A(T_{t+k})|T_t]\Big)^{2/p+1} && \text{By Jensen's Inequality} \\
&= (1-\delta)\cdot p\sigma_{min}(B_S)^2 \cdot \frac{(\Phi_A(S) - E_k)^{2/p+1}}{16k\Phi_A(S)^{2/p}}
\end{aligned}$$

Now, suppose that $\Delta_i \geq \Phi_A(S) - E_s \geq \Delta_{i+1}$, for $s = 0, \ldots, t'-1$. Then, for all such s, $E_{s+1} - E_s \geq \frac{(1-\delta)p\sigma_{min}(B_S)^2\Delta_{i+1}^{2/p+1}}{16kF^{2/p}}$. Summing these inequalities for $s = 0, \ldots, t'-1$, we find that

$$E_{t'} - E_0 \geq \frac{(1-\delta)p\sigma_{min}(B_S)^2}{16kF^{2/p}} \cdot \Delta_{i+1}^{2/p+1} \cdot t'$$

and for the increase from E_0 to $E_{t'}$ to be greater than Δ_{i+1}, it suffices to have

$$t' \geq \frac{32kF^{2/p}}{\Delta_{i+1}^{2/p} \cdot (1-\delta)p\sigma_{min}(B_S)^2}$$

In summary, if $\Phi_A(S) - \mathbb{E}[\Phi_A(T_t)] \leq \Delta_i$, then in at most $s = \frac{32kF^{2/p}}{\Delta_{i+1}^{2/p}\cdot(1-\delta)p\sigma_{min}(B_S)^2}$ iterations, $\Phi_A(S) - \mathbb{E}[\Phi_A(T_{t+s})] \leq \Delta_{i+1}$. Thus, if we let $N \in \mathbb{N}$ such that $\Delta_{N+1} \leq \frac{\varepsilon}{(1-\delta)^{p/2}}F \leq \Delta_N$, then the number of iterations t needed to have $\Phi_A(S) - \mathbb{E}[\Phi_A(T_t)] < \Delta_{N+1}$ is at most

$$\begin{aligned}
\sum_{i=0}^{N} \frac{32kF^{2/p}}{\Delta_{i+1}^{2/p}\cdot(1-\delta)p\sigma_{min}(B_S)^2} &= \frac{32kF^{2/p}}{(1-\delta)p\sigma_{min}(B_S)^2}\sum_{i=0}^{N}\frac{1}{\Delta_{i+1}^{2/p}} \\
&= \frac{32kF^{2/p}}{(1-\delta)p\sigma_{min}(B_S)^2}\sum_{i=0}^{N}\frac{1}{4^{(N-i)/p}}\cdot\frac{1}{\Delta_{N+1}^{2/p}} \\
&\leq \frac{32kF^{2/p}}{(1-\delta)p\sigma_{min}(B_S)^2}\cdot\frac{4^{1/p}(1-\delta)}{\varepsilon^{2/p}F^{2/p}}\sum_{i=0}^{N}\frac{1}{4^{(N-i)/p}} && \text{Since } \Delta_{N+1}^{2/p} \geq (\frac{\varepsilon F}{2(1-\delta)^{p/2}})^{2/p} \\
&\leq \frac{128k}{p\sigma_{min}(B_S)^2\varepsilon^{2/p}}\sum_{i=0}^{N}\frac{1}{2^i} && \text{Since } p < 2 \\
&\leq \frac{256k}{p\sigma_{min}(B_S)^2\varepsilon^{2/p}}
\end{aligned}$$

Thus, after $t = O(\frac{k}{p\sigma_{min}(B_S)^2\varepsilon^{2/p}})$ iterations,

$$\Phi_A(S) - \mathbb{E}[\Phi_A(T_t)] \leq \frac{\varepsilon}{(1-\delta)^{p/2}}\Phi_A(S) \leq \frac{\varepsilon}{\sqrt{1-\delta}}\Phi_A(S)$$

meaning

$$\|A\|_{p,2}^p - \|A - \pi_S A\|_{p,2}^p - \mathbb{E}[\|A\|_{p,2}^p - \|A - \pi_T A\|_{p,2}^p] \leq \frac{\varepsilon}{\sqrt{1-\delta}}\|A\|_{p,2}^p - \frac{\varepsilon}{\sqrt{1-\delta}}\|A - \pi_S A\|_{p,2}^p$$

and rearranging gives

$$\mathbb{E}[\|A - \pi_T A\|_{p,2}^p] \leq \Big(1 - \frac{\varepsilon}{\sqrt{1-\delta}}\Big)\|A - \pi_S A\|_{p,2}^p + \frac{\varepsilon}{\sqrt{1-\delta}}\|A\|_{p,2}^p$$

and observe that if we select $\delta = \varepsilon$, then $\frac{1}{\sqrt{1-\delta}} = O(1)$ for $\varepsilon < \frac{1}{2}$. Therefore,

$$\begin{aligned} E[\|A - \pi_T A\|_{p,2}] &\leq E[\|A - \pi_T A\|_{p,2}^p]^{1/p} && \text{(By Jensen's inequality since } x^{1/p} \text{ is concave)} \\ &\leq \Big((1 - O(\varepsilon))\|A - \pi_S A\|_{p,2}^p + O(\varepsilon)\|A\|_{p,2}^p\Big)^{1/p} \\ &\leq (1 - O(\varepsilon))^{1/p}\|A - \pi_S A\|_{p,2} + O(\varepsilon)^{1/p}\|A\|_{p,2} && (x+y)^{1/p} \leq x^{1/p} + y^{1/p} \\ &\leq \|A - \pi_S A\|_{p,2} + O(\varepsilon)^{1/p}\|A\|_{p,2} \end{aligned}$$

In summary, if we select $O(\frac{k}{p\sigma_{min}(B_S)^2\varepsilon^{2/p}})$ columns, then

$$E[\|A - \pi_T A\|_{p,2}] \leq \|A - \pi_S A\|_{p,2} + O(\varepsilon)^{1/p}\|A\|_{p,2}$$

and replacing ε with $O(\varepsilon^p)$, we find that

$$E[\|A - \pi_T A\|_{p,2}] \leq \|A - \pi_S A\|_{p,2} + \varepsilon\|A\|_{p,2}$$

after $O(\frac{k}{p\sigma_{min}(B_S)^2\varepsilon^2})$ iterations.

Running Time. Each evaluation of the error $\min_V \|A_{T\cup j}V - A\|_{p,2}$ takes $O(d|T|^2 + nd|T|)$ time by taking the pseudo-inverse of $A_{T\cup j}$. Since the algorithm samples $\frac{n}{k}\log(\frac{1}{\delta})$ columns at each iteration, the time it takes for each iteration is $O(\frac{n}{k}\log(\frac{1}{\delta})(d|T|^2 + nd|T|))$. If we set the number of iterations, i.e. the number of selected columns $r = \max|T| = \frac{k}{p\sigma^2\epsilon^2}$, the overall running time is then $O(\frac{k}{p\sigma^2\epsilon^2} \cdot \frac{n}{k}\log(\frac{1}{\delta}) \cdot (\frac{dk^2}{p^2\sigma^4\epsilon^4} + \frac{ndk}{p\sigma^2\epsilon^2})) = O(\frac{n}{p\sigma^2\epsilon^2}\log(\frac{1}{\delta}) \cdot (\frac{dk^2}{p^2\sigma^4\epsilon^4} + \frac{ndk}{p\sigma^2\epsilon^2}))$.

□

G Additional Experimental Details

Hyperparameters for k-CSS$_{1,2}$. There are two additional hyperparameters for our $O(1)$-approximation bi-criteria k-CSS$_{1,2}$ (see Section B.3 for a complete description of this algorithm), i.e. the size of the sparse embedding matrix and its sparsity, which we use to generate a rank-k left factor that gives an $O(1)$-approximation (see Section B.3.1 for details on the embedding matrix and sparsity). In the experiments, we set both the sparsity and the size of the sketching matrix we use to be $\frac{k}{2}$, where k is the number of output columns.

References

[1] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 91–100. ACM, 2013.

[2] Zhao Song, David P. Woodruff, and Peilin Zhong. Low rank approximation with entrywise l1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 688–701, New York, NY, USA, 2017. Association for Computing Machinery.

[3] Michael B. Cohen and Richard Peng. Lp row sampling by lewis weights. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 183–192, New York, NY, USA, 2015. Association for Computing Machinery.

[4] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

[5] Kenneth Clarkson, Ruosong Wang, and David Woodruff. Dimensionality reduction for tukey regression. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1262–1271. PMLR, 09–15 Jun 2019.

[6] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, page 932–941, USA, 2008. Society for Industrial and Applied Mathematics.

[7] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. Comput.*, 38(5):2060–2078, 2009.

[8] J. Bourgain, J. Lindenstrauss, and V. Milman. Approximation of zonoids by zonotopes. *Acta Mathematica*, 162(none):73 – 141, 1989.

[9] Christian Sohler and David P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 802–813. IEEE Computer Society, 2018.

[10] Kenneth L. Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, page 310–329, USA, 2015. IEEE Computer Society.

[11] Jelani Nelson and Huy L. Nguyen. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, page 117–126, USA, 2013. IEEE Computer Society.

[12] Grigoris Paouris, Petros Valettas, and Joel Zinn. Random version of dvoretzky's theorem in ℓ_p^n. *Stochastic Processes and their Applications*, 127(10):3187 – 3227, 2017.

[13] Andrew McGregor. Graph stream algorithms: A survey. *SIGMOD Rec.*, 43(1):9–20, May 2014.

[14] Ruosong Wang and David P. Woodruff. Tight bounds for lp oblivious subspace embeddings. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, page 1825–1843, USA, 2019. Society for Industrial and Applied Mathematics.

[15] Jiyan Yang, Yin-Lam Chow, Christopher Ré, and Michael W. Mahoney. Weighted sgd for ℓ_p regression with randomized preconditioning. *Journal of Machine Learning Research*, 18(211):1–43, 2018.

[16] Flavio Chierichetti, Screenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P. Woodruff. Algorithms for ℓ_p Low-Rank Approximation. 2017.

[17] Arvind V Mahankali and David P Woodruff. Optimal ℓ_1 column subset selection and a fast ptas for low rank approximation. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 560–578. SIAM, 2021.

[18] Jason Altschuler, Aditya Bhaskara, Gang Fu, Vahab Mirrokni, Afshin Rostamizadeh, and Morteza Zadimoghaddam. Greedy column subset selection: New bounds and distributed

algorithms. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2539–2548. JMLR.org, 2016.

[19] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 1812–1818. AAAI Press, 2015.