Near-Optimal Column-Based Matrix Reconstruction

Christos Boutsidis†  Petros Drineas‡  Malik Magdon-Ismail§

Abstract
We consider low-rank reconstruction of a matrix using a subset of its columns and we present asymptotically optimal algorithms for both spectral norm and Frobenius norm reconstruction. The main tools we introduce to obtain our results are: (i) the use of fast approximate SVD-like decompositions for column-based matrix reconstruction, and (ii) two deterministic algorithms for selecting rows from matrices with orthonormal columns, building upon the sparse representation theorem for decompositions of the identity that appeared in [1].

1 Introduction
The best rank $k$ approximation to a matrix $A \in \mathbb{R}^{m \times n}$ is $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq 0$ are the top $k$ singular values of $A$, with associated left and right singular vectors $u_i \in \mathbb{R}^m$ and $v_i \in \mathbb{R}^n$ respectively. (See Section 1.1 for notation.) The singular values and singular vectors of $A$ can be computed via the Singular Value Decomposition (SVD) of $A$ in $O(mn \min\{m,n\})$ time.

There is considerable interest (e.g. [4, 6, 8, 9, 11, 15, 19, 20, 21]) in determining a minimum set of $r \ll n$ columns of $A$ which is approximately as good as $A_k$ at reconstructing $A$. Such columns are important for interpreting data [21], building robust machine learning algorithms [4], etc.

Let $A \in \mathbb{R}^{m \times n}$ and let $C \in \mathbb{R}^{m \times r}$ consist of $r$ columns of $A$ for some $k \leq r < n$. We are interested in the reconstruction errors

$$\|A - CC^+ A\|_\xi \quad \text{and} \quad \|A - \Pi_{C,k}^\xi(A)\|_\xi,$$

for $\xi = 2, F$ (see Section 1.1 for notation). The former is the reconstruction error for $A$ using the columns in $C$; the latter is the error from the best rank $k$ reconstruction of $A$ (under the appropriate norm) within the column space of $C$. For fixed $A$, $k$, and $r$, we would like these errors to be as close to

$$\|A - A_k\|_\xi$$

as possible. We present polynomial-time near-optimal constructions for arbitrary $r > k$, settling important open questions regarding column-based matrix reconstruction.

- **Spectral norm:** What is the best reconstruction error with $r > k$ columns? We present polynomial-time (deterministic and randomized) algorithms with approximation error asymptotically matching a lower bound proven in this work. Prior work had focused on the $r = k$ case and presented near-optimal polynomial-time algorithms [6, 17].

- **Frobenius norm:** How many columns are needed for relative error approximation, i.e. a reconstruction error of $(1 + \epsilon)\|A - A_k\|_F$, for $\epsilon > 0$? We show that $O(k/\epsilon)$ columns contain a rank-$k$ subspace which reconstructs $A$ to relative error, and we present the first sub-SVD...
general, the above theorem matches a lower bound that we prove in Section 9.1. This is the first spectral reconstruction algorithm with asymptotically optimal guarantees for arbitrary $\|\minimizes\ A_{i,j} A_{i,j}^2$ and $\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2$; $\|A\|_2$ is used if a result holds for both norms $\xi = 2$ and $\xi = F$. The Singular Value Decomposition (SVD) of $A$, with rank($A$) = $\rho$ is

$$A = \begin{pmatrix} U_k & U_{\rho-k} \end{pmatrix} \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_{\rho-k} \end{pmatrix} \begin{pmatrix} V_k^T \\ V_{\rho-k}^T \end{pmatrix},$$

with singular values $\sigma_1 \geq \ldots \sigma_k \geq \sigma_{k+1} \geq \ldots \geq \sigma_0 > 0$. We will use $\sigma_i(A)$ to denote the $i$-th singular value of $A$ when the matrix is not clear from the context. The matrices $U_k \in \mathbb{R}^{m \times k}$ and $U_{\rho-k} \in \mathbb{R}^{m \times (\rho-k)}$ contain the left singular vectors of $A$, and, similarly, the matrices $V_k \in \mathbb{R}^{n \times k}$ and $V_{\rho-k} \in \mathbb{R}^{n \times (\rho-k)}$ contain the right singular vectors of $A$. It is well-known that $A_k = U_k \Sigma_k V_k^T$ minimizes $\|A - X\|_{\xi}$ over all matrices $X \in \mathbb{R}^{m \times n}$ of rank at most $k$. We use $A_{\rho-k}$ to denote the matrix $A - A_k = U_{\rho-k} \Sigma_{\rho-k} V_{\rho-k}^T$. Also, $A^+ = V_A A^{-1} U_A^T$ denotes the Moore-Penrose pseudo-inverse of $A$. For a symmetric positive definite matrix $A = B B^T$, $\lambda_i(A) = \sigma_i^2(B)$ denotes the $i$-th eigenvalue of $A$.

Finally, given a matrix $A \in \mathbb{R}^{m \times n}$ and a matrix $C \in \mathbb{R}^{m \times r}$ with $r > k$, we formally define the matrix $I_{C,k}(A) \in \mathbb{R}^{m \times n}$ as the best approximation to $A$ within the column space of $C$ that has rank at most $k$. $I_{C,k}(A)$ minimizes $\|A - \hat{A}\|_{\xi}$, over all $\hat{A}$ in the column space of $C$ that have rank at most $k$ (one can write $I_{C,k}(A) = CX$ where $X \in \mathbb{R}^{r \times n}$ has rank at most $k$). In general, $I_{C,k}(A) \neq I_{C,k}(A)$; Section 2.2 discusses the computation of $I_{C,k}(A)$.

1.2 Our main results

Since $\|A - CC^+A\|_{F} \leq \|A - I_{C,k}(A)\|_{F}$, we will state all our bounds in terms of the latter quantity. Note that we chose to state our Frobenius norm bounds in terms of the square of the Frobenius norm; this choice facilitates comparisons with prior work and simplifies our proofs.

**Theorem 1** (Deterministic spectral norm reconstruction). Given $A \in \mathbb{R}^{m \times n}$ of rank $\rho$ and a target rank $k < \rho$, there exists a deterministic polynomial-time algorithm to select $r > k$ columns of $A$ and form a matrix $C \in \mathbb{R}^{m \times r}$ such that

$$\|A - I_{C,k}(A)\|_2 \leq \left(1 + \frac{1 + \sqrt{(\rho-k)/r}}{1 - \sqrt{k/r}}\right) \|A - A_k\|_2 = O\left(\sqrt{\rho/r}\right) \|A - A_k\|_2.$$ 

The matrix $C$ can be computed in $T_{SVD} + O\left(r m (k^2 + (\rho-k)^2)\right)$ time, where $T_{SVD}$ is the time needed to compute all $\rho$ right singular vectors of $A$.

Our algorithm uses the matrices $V_k$ and $V_{\rho-k}$ of the right singular vectors of $A$. These matrices can be computed in $O(mn \min\{m, n\})$ time via the SVD. The asymptotic multiplicative error of the above theorem matches a lower bound that we prove in Section 9.1. This is the first spectral reconstruction algorithm with asymptotically optimal guarantees for arbitrary $r > k$. Previous
work presented near-optimal algorithms for \( r = k \) [17]. We note that in Section 4 we will present a result that achieves a slightly worse error bound (essentially replacing \( \rho \) by \( n \) in the accuracy guarantee), but only uses the top \( k \) right singular vectors of \( A \) (i.e., the matrix \( V_k \)).

**Theorem 2** (Deterministic Frobenius norm reconstruction). Given \( A \in \mathbb{R}^{m \times n} \) of rank \( \rho \) and a target rank \( k < \rho \), there exists a deterministic polynomial-time algorithm to select \( r > k \) columns of \( A \) and form a matrix \( C \in \mathbb{R}^{m \times r} \) such that

\[
\|A - \Pi_{C,k}^F(A)\|_F^2 \leq \left( 1 + \frac{1}{1 - \sqrt{k/r}} \right) \|A - A_k\|_F^2.
\]

The matrix \( C \) can be computed in \( T_{V_k} + O(mn + nrk^2) \) time, where \( T_{V_k} \) is the time needed to compute the top \( k \) right singular vectors of \( A \).

Our bound implies a constant-factor approximation. Previous work presents deterministic near-optimal algorithms for \( r = k \) [6]; we are unaware of any deterministic algorithms for \( r > k \).

The next two theorems guarantee (up to small constant factors) the same bounds as Theorems 1 and 2, but the proposed algorithms are considerably more efficient. In particular, there is no need to exactly compute the right singular vectors of \( A \), because approximations suffice.

**Theorem 3** (Fast spectral norm reconstruction). Given \( A \in \mathbb{R}^{m \times n} \) of rank \( \rho \), a target rank \( 2 \leq k < \rho \), and \( 0 < \epsilon < 1 \), there exists a randomized algorithm to select \( r > k \) columns of \( A \) and form a matrix \( C \in \mathbb{R}^{m \times r} \) such that

\[
\mathbb{E} \left[ \|A - \Pi_{C,k}^2(A)\|_2 \right] \leq \left( \sqrt{2} + \epsilon \right) \left( 1 + \frac{1 + \sqrt{n/r}}{1 - \sqrt{k/r}} \right) \|A - A_k\|_2 = O \left( \sqrt{n/r} \right) \|A - A_k\|_2.
\]

The matrix \( C \) can be computed in \( O(mn\epsilon^{-1} \log(k^{-1} \min\{m,n\}) + nrk^2) \) time.

**Theorem 4** (Fast Frobenius norm reconstruction). Given \( A \in \mathbb{R}^{m \times n} \) of rank \( \rho \), a target rank \( 2 \leq k < \rho \), and \( 0 < \epsilon < 1 \), there exists a randomized algorithm to select \( r > k \) columns of \( A \) and form a matrix \( C \in \mathbb{R}^{m \times r} \) such that

\[
\mathbb{E} \left[ \|A - \Pi_{C,k}^F(A)\|_F^2 \right] \leq (1 + \epsilon) \left( 1 + \frac{1}{(1 - \sqrt{k/r})^2} \right) \|A - A_k\|_F^2.
\]

The matrix \( C \) can be computed in \( O(mn\epsilon^{-1} + nrk^2) \) time.

Our last, yet perhaps most interesting result, guarantees relative-error Frobenius norm approximation by combining the algorithm of Theorem 4 with one round of adaptive sampling [8, 9]. This is the first relative-error approximation for Frobenius norm reconstruction that uses a linear number of columns in \( k \) (the target rank). Previous work [12, 23, 8, 6] achieves relative error with \( O(k \log k + k/\epsilon) \) columns. Our result asymptotically matches the \( \Omega(k/\epsilon) \) lower bound in [8].

**Theorem 5** (Fast relative-error Frobenius norm reconstruction). Given \( A \in \mathbb{R}^{m \times n} \) of rank \( \rho \), a target rank \( 2 \leq k < \rho \), and \( 0 < \epsilon < 1 \), there exists a randomized algorithm to select at most

\[
r = \frac{2k}{\epsilon} (1 + o(1))
\]

columns of \( A \) and form a matrix \( C \in \mathbb{R}^{m \times r} \) such that

\[
\mathbb{E} \left[ \|A - \Pi_{C,k}^F(A)\|_F^2 \right] \leq (1 + \epsilon) \|A - A_k\|_F^2.
\]

The matrix \( C \) can be computed in \( O((mnk + nk^3) \epsilon^{-2/3}) \) time.
Running times. Our running times are stated in terms of the number of operations needed to compute the matrix $C$, and, for simplicity, we assume that $A$ is dense; if $A$ is sparse, additional savings might be possible. Our accuracy guarantees are in terms of the optimal matrix $\Pi_{C,k}(A)$, which would require additional time to compute. For the Frobenius norm, computing $\Pi_{C,k}(A)$ is straightforward, and only requires an additional $O(mnr + (m + n)r^2)$ time (see the discussion in Section 2.2). For the spectral norm, we are not aware of any algorithm to compute $\Pi_{k,r}^2(A)$ exactly. In Section 2.2 we present a simple approach that computes $\hat{\Pi}_{C,k}(A)$, a constant-factor approximation to $\Pi_{C,k}(A)$, in $O(mnr + (m + n)r^2)$ time. Our bounds in Theorems 1 and 3 can be restated in terms of the error $\|A - \Pi_{C,k}(A)\|_2^2$; the accuracy guarantees only weaken by small constant factors.

1.3 Lower Bounds

Table 1 provides a summary on lower bounds for the ratio

$$\frac{\|A - \Pi_{C,k}^\xi(A)\|_2^\xi}{\|A - A_k\|_2^\xi}.$$

where $C$ is a matrix consisting of $r$ columns of $A$, with $r \geq k$. Theorem 34 contributes a new lower bound for the spectral norm case when $r > k$. Note that any lower bound for the ratio $\|A - CC^+A\|_2^2/\|A - A_k\|_2^2$ implies the same lower bound for $\|A - \Pi_{C,k}^\xi(A)\|_2^\xi/\|A - A_k\|_2^\xi$; the converse, however, is not true.

1.4 Prior results on column-based matrix reconstructions

There is a long literature on algorithms for column-based matrix reconstruction using $r \geq k$ columns. The first result goes back to [16], with the most recent one being, to the best of our knowledge, the work in [6].

1.4.1 The Frobenius norm case

We present known upper bounds for the approximation ratio $\|A - \Pi_{C,k}^F(A)\|_F^2/\|A - A_k\|_F^2$. We start with the $r = k$ case. [3] describes a $T_{SV} + O(nk + k^2 \log^2 k \log \log k)$ time randomized algorithm which provides an upper bound $O(k \log^2 k)$ with constant probability. This bound was subsequently improved in [6]. More precisely, Theorem 8 of [6] gives a $(k + 1)$ deterministic approximation running in $O(knm^3 \log m)$ time; this upper bound matches the lower bound in [9]. [6] also presents three randomized algorithms such that $E\left[\|A - \Pi_{C,k}^F(A)\|_F^2\right] = (k + 1)\|A - A_k\|_F^2$. These randomized algorithms are presented in Theorem 7, Proposition 16, and Proposition 18 and run in $O(knm^3 \log m)$, $O(kn^3m + kn^4 \log n)$, and $O(kT_{SVD} + kmn^2)$ time, respectively. Moreover, Theorem 9 in [6] presents an $O(mn \log nk^2 \epsilon e^{-2} + n \log^3 n \cdot k^5 \epsilon e^{-6} \log (ke^{-1} \log n))$ time randomized algorithm such that, with constant probability, $\|A - \Pi_{C,k}^F(A)\|_F^2 \leq (1 + \epsilon) \cdot (k + 1)\|A - A_k\|_F^2$, for any $0 < \epsilon < 1$. Finally, [18] improved upon the running time of the results in [6]. More precisely,
Theorem 2 in [18] gives an $O(knm^2)$ time randomized algorithm with a $(k+1)$ multiplicative error (in expectation).

When $r = \Omega(k \log k)$, relative-error approximations are known. [12] presented the first result that achieved such a bound, using random sampling of the columns of $A$ according to the Euclidean norms of the rows of $V_k$. More specifically, a $(1+\epsilon)$-approximation was proven using $r = \Omega\left(k \epsilon^{-2} \log (k \epsilon^{-1})\right)$ columns in $T_{V_k} + O(kn + r \log r)$ time. [23] argued that the same technique gives a $(1+\epsilon)$-approximation using $r = \Omega\left(k \log k + k \epsilon^{-1}\right)$ columns. It also showed how to improve the running time to $T_{V_k} + O(kn + r \log r)$, where $\tilde{V}_k \in \mathbb{R}^{n \times k}$ contains the right singular vectors of an approximation to $A_k$ and can be computed in $o(nm \min\{m,n\})$ time, which is less than the time needed to compute the SVD of $A$. In [8], the authors leveraged volume sampling and presented an approach that achieves a relative error approximation using $O(k^2 \log k + k \epsilon^{-1})$ columns in $O(mnk^2 \log k)$ time. Also, it is possible to combine the fast volume sampling approach in [6] (setting, for example, $\epsilon = 1/2$) with $O(\log k)$ rounds of adaptive sampling as described in [8] to achieve a relative error approximation using $O(k \log k + k \epsilon^{-1})$ columns. The running time of this combined algorithm is $O\left(mnk^2 \log n + nk^2 \log^2 n \cdot \log (k \log n)\right)$. The techniques in [12] do not apply to general $r > k$, since $\Omega(k \log k)$ columns must be sampled in order to preserve rank with random sampling.

A related line of work (including [7, 13, 14, 24]) has focused on the construction of coresets and sketches for high dimensional subspace approximation with respect to general $\ell_p$ norms. In our setting, $p = 2$ corresponds to Frobenius norm matrix reconstruction, and Theorem 1.3 of [24] presents an exponential in $k/\epsilon$ algorithm to select $O\left(k^2 \epsilon^{-1} \log (k/\epsilon)\right)$ columns that guarantee a relative error approximation. It would be interesting to understand if the techniques of [7, 13, 14, 24] can be extended to match our results here in the special case of $p = 2$.

The recent work in [18] presents a deterministic and a randomized algorithm for arbitrary $r \geq k$ that guarantee upper bounds for the ratio $\|A - CC^+A\|_F^2/\|A - A_k\|_F^2$. More precisely, Theorem 1 in [18] presents an $O(rnm^3 \log m)$ time deterministic algorithm with bound $(r+1)/(r+1-k)$, which is tight up to low order terms if $r = o(n)$. Also, Theorem 2 in [18] presents an $O(rnm^2)$ time randomized algorithm which achieves the same bound in expectation. We should notice that it is not obvious how to extend the results in [18] to obtain comparable bounds for the ratio $\|A - \Pi_{C,k}^F(A)\|_F^2/\|A - A_k\|_F^2$.

### 1.4.2 The spectral norm case

We present known guarantees for the approximation ratio $\|A - \Pi_{C,k}^F(A)\|_2^2/\|A - A_k\|_2^2$. In general, results for spectral norm have been sparse. When $r = k$, the strongest bound emerges from Strong Rank Revealing QR (RRQR) [17] (specifically Algorithm 4 in [17]), which, for $f > 1$, runs in $O(mnk \log f n)$ time and guarantees an $f^2 k(n - k) + 1$ approximation. For $r > k$, to the best of our knowledge, there is no easy way to extend the RRQR guarantees. In fact we are only aware of one bound that is applicable to this domain, other than those obtained by trivially extending the Frobenius norm bounds, because any $\alpha$-approximation in the Frobenius norm gives an $\alpha(\rho - k)$-approximation in the spectral norm:

$$\|A - \Pi_{C,k}^F(A)\|_2^2 \leq \|A - \Pi_{C,k}^F(A)\|_2^2 \leq \|A - \Pi_{C,k}^F(A)\|_2^2 \leq \alpha \|A - A_k\|_2^2 \leq \alpha(\rho - k) \|A - A_k\|_2^2.$$

That exception is the recent work of [2], which describes a deterministic $T_{V_k} + O\left(nk(n - r)\right)$ time algorithm that guarantees approximation error $2 + k(n - r)/(r - k + 1)$ for any $r \geq k$. 


2 Matrix norm properties and the computation of $\Pi_{C,k}^\xi(A)$

2.1 Matrix norm properties

Recall notation from Section 1.1; for any matrix $A$ of rank at most $\rho$, it is well-known that $\|A\|^2_F = \sum_{i=1}^\rho \sigma_i^2(A)$ and $\|A\|^2_2 = \sigma_1(A)$. Also, the best rank $k$ approximation to $A$ satisfies $\|A - A_k\|_2 = \sigma_{k+1}(A)$ and $\|A - A_k\|^2_F = \sum_{i=k+1}^\rho \sigma_i^2(A)$. For any two matrices $A$ and $B$ of appropriate dimensions, $\|A\|_2 \leq \|A\|_F \leq \sqrt{\rho} \|A\|_2$, $\|AB\|_F \leq \|A\|_F \|B\|_2$, and $\|AB\|_F \leq \|A\|_2 \|B\|_F$. The latter two properties are stronger versions of the standard submultiplicativity property.

We refer to the next lemma as matrix-Pythagoras:

Lemma 6. If $X, Y \in \mathbb{R}^{m \times n}$ and $XY^T = 0_{m \times m}$ or $X^T Y = 0_{n \times n}$, then

$$\|X + Y\|^2_F = \|X\|^2_F + \|Y\|^2_F,$$

max\{\|X\|^2_F, \|Y\|^2_F\} \leq \|X + Y\|^2_F \leq \|X\|^2_F + \|Y\|^2_F.

Proof. Since $XY^T = 0_{m \times m}$, $(X + Y)(X + Y)^T = XX^T + YY^T$. For $\xi = F$, $\|X + Y\|^2_F = Tr((X + Y)(X + Y)^T) = Tr(XX^T + YY^T) = \|X\|^2_F + \|Y\|^2_F$.

Let $z$ be any vector in $\mathbb{R}^n$. For $\xi = 2$,

$$\|X + Y\|^2_2 = \max_{\|z\|^2_2=1} z^T (X + Y)(X + Y)^T z = \max_{\|z\|^2_2=1} (z^T XX^T z + z^T YY^T z).$$

We have that $\max_{\|z\|^2_2=1} (z^T XX^T z + z^T YY^T z)$ is at most

$$\max_{\|z\|^2_2=1} z^T XX^T z + \max_{\|z\|^2_2=1} z^T YY^T z = \|X\|^2_2 + \|Y\|^2_2$$

and that

$$\max_{\|z\|^2_2=1} (z^T XX^T z + z^T YY^T z) \geq \max_{\|z\|^2_2=1} z^T XX^T z = \|X\|^2_2,$$

since $z^T YY^T z$ is non-negative for any vector $z$. We get the same lower bound with $\|Y\|^2_2$ instead, which means we can lower bound with max\{\|X\|^2_2, \|Y\|^2_2\}. The case with $X^T Y = 0_{n \times n}$ can be proven similarly.

2.2 Computing the best rank $k$ approximation $\Pi_{C,k}^\xi(A)$

Let $A \in \mathbb{R}^{m \times n}$, let $k < n$ be an integer, and let $C \in \mathbb{R}^{m \times r}$ with $r > k$. Recall that $\Pi_{C,k}^\xi(A) \in \mathbb{R}^{m \times n}$ is the best rank $k$ approximation to $A$ in the column space of $C$: we can write $\Pi_{C,k}^\xi(A) = CX^\xi$, where

$$X^\xi = \arg\min_{\Psi \in \mathbb{R}^{r \times n}, \text{rank}(\Psi) \leq k} \|A - C\Psi\|^2_\xi.$$

In order to compute (or approximate) $\Pi_{C,k}^\xi(A)$ given $A$, $C$, and $k$, we will use the following algorithm:

1: Orthonormalize the columns of $C$ in $O(mr^2)$ time to construct the matrix $Q \in \mathbb{R}^{m \times r}$.
2: Compute $(Q^T A)_k \in \mathbb{R}^{r \times n}$ via SVD in $O(mnr + nr^2)$ – the best rank-$k$ approximation of $Q^T A$.
3: Return $Q (Q^T A)_k \in \mathbb{R}^{m \times n}$ in $O(mnk)$ time.

Clearly, $Q (Q^T A)_k$ is a rank $k$ matrix that lies in the column span of $C$. Note that though $\Pi_{C,k}^\xi(A)$ can depend on $\xi$, our algorithm computes the same matrix, independent of $\xi$. The next lemma, which is essentially Lemma 4.3 in [5] together with a slight improvment of Theorem 9.3 in [19], proves that this algorithm computes $\Pi_{C,k}^\xi(A)$ and a constant factor approximation to $\Pi_{C,k}^\xi(A)$. 

6
Lemma 7. Given $A \in \mathbb{R}^{m \times n}$, $C \in \mathbb{R}^{m \times r}$ and an integer $k$, the matrix $Q \left( Q^T A \right)_k \in \mathbb{R}^{m \times n}$ described above (where $Q$ is an orthonormal basis for the columns of $C$) can be computed in $O(mnr + (m + n)r^2)$ time and satisfies:

$$
\|A - Q \left( Q^T A \right)_k \|_F^2 = \|A - \Pi_{C,k}^F(A)\|_F^2,
$$

$$
\|A - Q \left( Q^T A \right)_k \|_2^2 \leq 2\|A - \Pi_{C,k}^2(A)\|_2^2.
$$

Proof. Our proof for the Frobenius norm case is a mild modification of the proof of Lemma 4.3 [5]. First, note that $\Pi_{C,k}^F(A) = \Pi_{Q,k}^F(A)$, because $Q \in \mathbb{R}^{m \times r}$ is an orthonormal basis for the column space of $C$. Thus,

$$
\|A - \Pi_{C,k}^F(A)\|_F^2 = \|A - \Pi_{Q,k}^F(A)\|_F^2 = \min_{\Psi: \text{rank}(\Psi) \leq k} \|A - Q\Psi\|_F^2.
$$

Now, using matrix-Pythagoras and the orthonormality of $Q$,

$$
\|A - Q\Psi\|_F^2 = \|A - QQ^TA + Q(Q^TA - \Psi)\|_F^2 = \|A - QQ^TA\|_F^2 + \|Q^TA - \Psi\|_F^2.
$$

Setting $\Psi = (Q^TA)_k$ minimizes the above quantity over all rank-$k$ matrices $\Psi$. Thus, combining the above results, $\|A - \Pi_{C,k}^F(A)\|_F^2 = \|A - Q \left( Q^T A \right)_k \|_F^2$.

We now proceed to the spectral-norm part of the proof, which combines ideas from Theorem 9.3 [19] and matrix-Pythagoras. We first manipulate the term $\|A - Q \left( Q^T A \right)_k \|_2^2$,

$$
\|A - Q \left( Q^T A \right)_k \|_2^2 = \|A - QQ^TA + Q(Q^TA - \Psi)\|_2^2
\leq \|A - QQ^TA\|_2^2 + \|Q^TA - \Psi\|_2^2
\leq \|A - \Pi_{Q,k}^2(A)\|_2^2 + \|A - A_k\|_2^2
\leq 2\|A - \Pi_{Q,k}^2(A)\|_2^2.
$$

The first inequality follows from the simple fact that $(QQ^TA)_k = Q \left( Q^T A \right)_k$ and matrix-Pythagoras; the first term in (a) follows because $QQ^TA$ is the (unconstrained, not necessarily of rank at most $k$) best approximation to $A$ in the column space of $Q$; the second term in (a) follows because $QQ^T$ is a projector matrix and thus

$$
\|Q^TA - (QQ^TA)_k\|_2^2 = \sigma_{k+1}^2(QQ^TA) \leq \sigma_{k+1}^2(A) = \|A - A_k\|_2^2.
$$

The last inequality follows because $\|A - A_k\|_2^2 \leq \|A - \Pi_{Q,k}^2(A)\|_2^2$.

3 Main Tools

Our two main tools are the use of matrix factorizations for column-based low-rank matrix reconstruction, and two deterministic sparsification lemmas which extend the work of [1].

3.1 Matrix factorizations

Our first tool suggests how to use matrix factorizations to reconstruct a matrix from a subset of its columns: Lemmas 8, 10, and 11. Lemmas 10 and 11 present factorizations of the matrix $A \in \mathbb{R}^{m \times n}$ of the form

$$
A = BZ^T + E,
$$

where $B \in \mathbb{R}^{m \times k}$, $Z \in \mathbb{R}^{n \times k}$, $E \in \mathbb{R}^{m \times n}$, and $Z$ consists of orthonormal columns. Lemma 8 shows how to apply these factorizations by drawing a connection between matrix factorizations and column selection. Lemma 8 is the starting point of all our column reconstruction results.
Lemma 8. Let \( A = BZ^T + \mathbf{E} \), with \( \mathbf{E} Z = 0_{m \times k} \) and \( Z^T Z = \mathbf{I}_k \). Let \( S \in \mathbb{R}^{n \times r} \) be any matrix such that \( \text{rank}(Z^T S) = \text{rank}(Z) = k \). Let \( C = AS \in \mathbb{R}^{m \times r} \). Then,

\[
\| A - \Pi^k_{C,k}(A) \|_\xi^2 \leq \| E \|_\xi^2 + \| ES(Z^T S)^+ \|_\xi^2.
\]

Proof. The optimality of \( \Pi^k_{C,k}(A) \) implies that \( \| A - \Pi^k_{C,k}(A) \|_\xi^2 \leq \| A - X \|_\xi^2 \) over all matrices \( X \in \mathbb{R}^{m \times n} \) of rank at most \( k \) in the column space of \( C \). Consider the matrix \( X = C(Z^T S)^+ Z^T \) (clearly \( X \) is in the column space of \( C \) and \( \text{rank}(X) \leq k \) because \( Z \in \mathbb{R}^{n \times k} \):

\[
\| A - C(Z^T S)^+ Z^T \|_\xi^2 = \| BZ^T + (A - BZ^T) \|_A = \| (BZ^T + E) S(Z^T S)^+ Z^T \|_\xi^2
\]

\[
= \| BZ^T - B(Z^T S)(Z^T S)^+ Z^T \|_\xi^2 + \| E + ES(Z^T S)^+ Z^T \|_\xi^2
\]

\[
= \| E \|_\xi^2 + \| ES(Z^T S)^+ Z^T \|_\xi^2
\]

\[
\leq \| E \|_\xi^2 + \| ES(Z^T S)^+ Z^T \|_\xi^2.
\]

(\( a \)) follows because, by assumption, \( \text{rank}(Z^T S) = k \), and thus \( (Z^T S)(Z^T S)^+ = I_k \) which implies \( BZ^T - B(Z^T S)(Z^T S)^+ Z^T = 0_{m \times n} \). (\( b \)) follows by matrix-Pythagoras because \( ES(Z^T S)^+ Z^T E^T = 0_{m \times n} \) (recall that \( E = A - BZ^T \) and \( \mathbf{E} Z = 0_{m \times k} \) by assumption). The lemma follows by strong submultiplicativity because \( Z \) has orthonormal columns, hence \( \| Z \|_2 = 1 \).

In this work, we view \( C \) as a dimensionally-reduced or sampled sketch of \( A \); \( S \) is the dimension-reduction or sampling matrix. In words, Lemma 8 argues that if the matrix \( S \) preserves the rank of an approximate factorization of the original matrix \( A \), then the reconstruction of \( A \) from \( C = AS \) has an error that is essentially proportional to the error of the approximate factorization. The importance of this lemma is that it indicates an algorithm for matrix reconstruction using a subset of the columns of \( A \): first, compute any factorization of the form \( A = BZ^T + E \) satisfying the assumptions of the lemma; then, compute a sampling matrix \( S \) which satisfies the rank assumption and controls the error \( \| ES(Z^T S)^+ \|_\xi^2 \).

An immediate corollary of Lemma 8 emerges by considering the SVD of \( A \). More specifically, consider the following factorization of \( A \): \( A = AV_k V_k^T + (A - A_k) \), where \( V_k \) is the matrix of the top \( k \) right singular vectors of \( A \). In the parlance of Lemma 8, \( Z = V_k, B = AV_k, E = A - A_k \), and clearly \( \mathbf{E} Z = 0_{m \times k} \).

Lemma 9. Let \( S \in \mathbb{R}^{n \times r} \) be a matrix such that \( \text{rank}(V_k^T S) = k \). Let \( C = AS \); then,

\[
\| A - \Pi^k_{C,k}(A) \|_\xi^2 \leq \| A - A_k \|_\xi^2 + \| (A - A_k) S(V_k^T S)^+ \|_\xi^2.
\]

The above lemma will be useful for designing the deterministic (spectral norm and Frobenius norm) column-reconstruction algorithms of Theorems 1 and 2. However, computing the SVD is costly and thus we would like to design a factorization of the form \( A = BZ^T + E \) that is as good as the SVD, but can be computed in \( O(mnk) \) time. The next two lemmas achieve this goal by extending the algorithms in \( [19, 22] \) (see Sections 5 and 6 for their proofs). We will use these factorizations to design fast column reconstruction algorithms in Theorems 3, 4, and 5.

Lemma 10 (Randomized fast spectral norm SVD). Given \( A \in \mathbb{R}^{m \times n} \) of rank \( \rho \), a target rank \( 2 \leq k < \rho \), and \( 0 < \epsilon < 1 \), there exists an algorithm that computes a factorization \( A = BZ^T + E \), with \( B = AZ, Z^T Z = \mathbf{I}_k \), and \( \mathbf{E} Z = 0_{m \times k} \) such that

\[
\mathbb{E} \| E \|_2 \leq (\sqrt{2} + \epsilon) \| A - A_k \|_2.
\]

The proposed algorithm runs in \( O(mnk^{1 - \epsilon} \log (k^{-1} \min\{m, n\})) \) time.
Lemma 11 (Randomized fast Frobenius norm SVD). Given $A \in \mathbb{R}^{m \times n}$ of rank $\rho$, a target rank $2 \leq k < \rho$, and $0 < \epsilon < 1$, there exists an algorithm that computes a factorization $A = BZ^T + E$, with $B = AZ$, $Z^TZ = I_k$, and $EZ = 0_{m \times k}$ such that $E \left\| E \right\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2$.

The proposed algorithm runs in $O(mnk\epsilon^{-1})$ time.

### 3.2 Sparse approximate decompositions of the identity

Lemmas 8, 10 and 11 argue that, in order to achieve almost optimal column-based matrix reconstruction, we need a sampling matrix $S$ that preserves the rank of $Z$ and controls the error $\|ES(Z^TS)^+\|_F$. We present algorithms to compute such a matrix $S$ in Lemmas 12 and 13. These lemmas were motivated by an important linear-algebraic result for a decomposition of the identity presented by Batson et al. [1]. It is worth emphasizing that the result of [1] cannot be directly applied to the column reconstruction problem. Indeed, in our setting, it is necessary to control properties related to both matrices $Z$ and $E = A - BZ^T$ simultaneously. In the spectral-norm reconstruction case, we need to control the singular values of the two matrices; in the Frobenius-norm reconstruction case, we need to control singular values and Frobenius norms of two matrices. The following two lemmas are proven in Sections 7 and 8.

**Lemma 12 (Dual Set Spectral Sparsification.).** Let $V = \{v_1, \ldots, v_n\}$ and $U = \{u_1, \ldots, u_n\}$ be two equal cardinality decompositions of the identity, where $v_i \in \mathbb{R}^k$ ($k < n$), $u_i \in \mathbb{R}^\ell$ ($\ell \leq n$), $\sum_{i=1}^n v_i v_i^T = I_k$, and $\sum_{i=1}^n u_i u_i^T = I_{\ell}$. Given an integer $r$ with $k < r \leq n$, there exists a set of weights $s_i \geq 0$ ($i = 1, \ldots, n$) at most $r$ of which are non-zero, such that

$$
\lambda_k \left( \sum_{i=1}^n s_i v_i v_i^T \right) \geq \left( 1 - \sqrt{\frac{k}{r}} \right)^2, \\
\lambda_1 \left( \sum_{i=1}^n s_i u_i u_i^T \right) \leq \left( 1 + \sqrt{\frac{\ell}{r}} \right)^2.
$$

The weights $s_i$ can be computed deterministically in $O(rn(k^2 + \ell^2))$ time.

In matrix notation, let $U$ and $V$ be the matrices whose rows are the vectors $u_i$ and $v_i$, respectively. We can now construct the sampling matrix $S \in \mathbb{R}^{n \times r}$ as follows: for $i = 1, \ldots, n$, if $s_i$ is non-zero then include $\sqrt{s_i}e_i$ as a column of $S$; here $e_i$ is the $i$-th standard basis vector. Using this matrix notation, the above lemma guarantees that $\sigma_k(V^T S) \geq 1 - \sqrt{k}/r$ and $\sigma_1(U^T S) \leq 1 + \sqrt{\ell}/r$. Clearly, $S$ may be viewed as a matrix that samples and rescales $r$ rows of $U$ and $V$ (columns of $U^T$ and $V^T$), namely the rows that correspond to non-zero weights $s_i$.

**Lemma 13 (Dual Set Spectral-Frobenius Sparsification.).** Let $V = \{v_1, \ldots, v_n\}$ be a decomposition of the identity, where $v_i \in \mathbb{R}^k$ ($k < n$) and $\sum_{i=1}^n v_i v_i^T = I_k$; let $A = \{a_1, \ldots, a_n\}$ be an arbitrary set of vectors, where $a_i \in \mathbb{R}^\ell$. Then, given an integer $r$ such that $k < r \leq n$, there exists a set of weights $s_i \geq 0$ ($i = 1 \ldots n$), at most $r$ of which are non-zero, such that

$$
\lambda_k \left( \sum_{i=1}^n s_i v_i v_i^T \right) \geq \left( 1 - \sqrt{\frac{k}{r}} \right)^2, \\
\text{Tr} \left( \sum_{i=1}^n s_i a_i a_i^T \right) \leq \text{Tr} \left( \sum_{i=1}^n a_i a_i^T \right) = \sum_{i=1}^n \|a_i\|_2^2.
$$

Note that we slightly abused notation: indeed, the number of columns of $S$ is less than or equal to $r$, since at most $r$ of the weights are non-zero. Here, we use $r$ to also denote the actual number of non-zero weights, which is equal to the number of columns of the matrix $S$. 


The weights $s_i$ can be computed deterministically in $O\left(rnk^2 + n\ell \right)$ time.

In matrix notation (here $A$ denotes the matrix whose rows are the vectors $a_i$), the above lemma guarantees that $\sigma_k(V^T S) \geq 1 - \sqrt{k/r}$ and $\|A^T S\|_F^2 \leq \|A\|_F^2$.

### 4 Proofs of our Main Results

In this section, we leverage the main tools described in Section 3 in order to prove the results of Section 1.2 (Theorems 1 through 5). We start with a proof of Theorem 1, using Lemmas 9 and 12.

**Proof of Theorem 1.** Apply the algorithm of Lemma 12 on the following two sets of vectors: the $n$ rows of the matrix $V_k$ and the $n$ rows of the matrix $V_{\rho-k}$. The output of the algorithm is a sampling and rescaling matrix $S \in \mathbb{R}^{n \times r}$ (see discussion after Lemma 12 in Section 3.2). Let $C = AS$ and note that $C$ consists of a subset of $r$ rescaled columns of $A$. Lemma 12 guarantees that $\sigma_k(V_k^T S) \geq 1 - \sqrt{k/r} > 0$ (assuming $r > k$), and so rank($V_k^T S$) = $k$. Also, $\sigma_1(V_{\rho-k}^T S) = \|V_{\rho-k}^T S\|_2 \leq 1 + \sqrt{(\rho-k)/r}$. Applying Lemma 9, we get

$$
\|A - \Pi_{C,k}^2(A)\|_2^2 \leq \|A - A_k\|_2^2 + \|(A - A_k)S(V_k^T S)^+\|_2^2
$$

$$
\leq \|A - A_k\|_2^2 + \|(A - A_k)S\|_2^2\|(V_k^T S)^+\|_2^2
$$

$$
\leq \|A - A_k\|_2^2 + \|\Sigma_{\rho-k}\|_2^2\|V_{\rho-k}^T S\|_2^2\|(V_k^T S)^+\|_2^2
$$

$$
\leq \|A - A_k\|_2^2 \left(1 + \frac{1 + \sqrt{(\rho-k)/r}^2}{(1 - \sqrt{k/r})^2}\right),
$$

where the last inequality follows because $\|\Sigma_{\rho-k}\|_2 = \|A - A_k\|_2$ and $\|(V_k^T S)^+\|_2 = 1/\sigma_k(V_k^T S) \leq 1/\left(1 - \sqrt{k/r}\right)$. Theorem 1 now follows by taking square roots of both sides and using $\sqrt{1 + x^2} \leq 1 + x$. The running time is equal to the time needed to compute $V_k$ and $V_{\rho-k}$ plus the running time of the algorithm in Lemma 12. Finally, we note that the rescaling of the columns of $C$ does not change the span of its columns and thus is irrelevant in the construction of $\Pi_{C,k}^2(A)$.

Our next theorem describes a deterministic algorithm for spectral norm reconstruction that only needs to compute $V_k$ and will serve as a prequel to the proof of Theorem 3. The accuracy guarantee of this theorem is essentially identical to the one in Theorem 1, with $\rho - k$ being replaced by $n$.

**Theorem 14.** Given $A \in \mathbb{R}^{m \times n}$ of rank $\rho$ and a target rank $k < \rho$, there exists a deterministic polynomial-time algorithm to select $r > k$ columns of $A$ and form a matrix $C \in \mathbb{R}^{m \times r}$ such that

$$
\|A - \Pi_{C,k}^2(A)\|_2 \leq \|A - A_k\|_2 + \left(\frac{1 + \sqrt{n/r}}{1 - \sqrt{k/r}}\right) \|A - A_k\|_2.
$$

The matrix $C$ can be computed in $T_{V_k} + O\left(nrk^2\right)$ time, where $T_{V_k}$ is the time needed to compute the top $k$ right singular vectors of $A$.

**Proof.** The proof is very similar to the proof of Theorem 1, so we only highlight the differences. First, apply the algorithm of Lemma 12 on the following two sets of vectors: the $n$ rows of the matrix $V_k$ and the $n$ rows of the matrix $I_n$. The output of the algorithm is a sampling and rescaling matrix $S \in \mathbb{R}^{n \times r}$ (see discussion after Lemma 12 in Section 3.2). Let $C = AS$ and note that $C$ consists of a subset of $r$ rescaled columns of $A$. Lemma 12 guarantees that $\|I_n S\|_2 \leq 1 + \sqrt{n/r}$. We now replicate the proof of Theorem 1 up to the point where $\|(A - A_k)S(V_k^T S)^+\|_2^2$ is bounded. We continue as follows:

$$
\|(A - A_k)S(V_k^T S)^+\|_2^2 = \|(A - A_k)I_n S(V_k^T S)^+\|_2^2
$$

$$
\leq \|A - A_k\|_2^2 \|I_n S\|_2^2 \|(V_k^T S)^+\|_2^2.
$$

10
Again, as in Theorem 1, the rescaling of the columns of \( C \) is irrelevant to the construction of \( \Pi_{C,k}^2(A) \). To analyze the running time of the proposed algorithm, we need to look more closely at Lemma 12 and the related algorithm. The proof of this Lemma in Section 7 argues that the algorithm of Lemma 12 can be implemented in \( O(nrk^2) \) time. The total running time is the time needed to compute \( V_k \) plus \( O(nrk^2) \).

**Proof of Theorem 3.** In order to prove Theorem 3 we will follow the proof of Theorem 1 using Lemma 10 (a fast matrix factorization) instead of Lemma 9 (the exact SVD of \( A \)). More specifically, instead of using the top \( k \) right singular vectors of \( A \) (the matrix \( V_k \)), we use the matrix \( Z \in \mathbb{R}^{n \times k} \) of Lemma 10. We now apply the algorithm of Lemma 12 on the following two sets of vectors: the \( n \) rows of the matrix \( Z \) and the \( n \) rows of the matrix \( I_n \). The output of the algorithm is a sampling and rescaling matrix \( S \in \mathbb{R}^{n \times r} \) (see discussion after Lemma 12 in Section 3.2). Let \( C = AS \) and note that \( C \) consists of a subset of \( r \) rescaled columns of \( A \). The proof of Theorem 3 is now identical to the proof of Theorem 14, except for using Lemma 8 instead of Lemma 9 in the first step of the proof:

\[
\|A - \Pi_{C,k}^2(A)\|_2^2 \leq \|E\|_2^2 + \|ES(Z^T S)^+\|_2^2 \\
= \|E\|_2^2 + \|EI_a S(Z^T S)^+\|_2^2 \\
\leq \|E\|_2^2 (1 + \|I_a S\|_2^2 \|Z^T S\|^+\|_2^2),
\]

where \( E \) is the residual error from the matrix factorization of Lemma 10. Taking square roots (using \( \sqrt{1 + x^2} \leq 1 + x \)) and using the bounds guaranteed by Lemma 12 for \( \|I_a S\|_2 \) and \( \|Z^T S\|^+\|_2 \), we obtain a bound in terms of \( \|E\|_2 \). Finally, since \( E \) is a random variable, taking expectations and applying the bound of Lemma 10 concludes the proof of the theorem. Again, the rescaling of the columns of \( C \) is irrelevant to the construction of \( \Pi_{C,k}^2(A) \). The running time is the time needed to compute the matrix \( Z \) from Lemma 10 plus an additional \( O(nrk^2) \) time as in Theorem 14.

**Proof of Theorem 2.** First, apply the algorithm of Lemma 13 on the following two sets of vectors: the \( n \) rows of the matrix \( V_k \) and the \( n \) rows of the matrix \( (A - A_k)^T \). The output of the algorithm is a sampling and rescaling matrix \( S \in \mathbb{R}^{n \times r} \) (see discussion after Lemma 12 in Section 3.2). Let \( C = AS \) and note that \( C \) consists of a subset of \( r \) rescaled columns of \( A \). We follow the proof of Theorem 1 in the previous section up to the point where we need to bound the term \( \|(A - A_k)S(V_k^T S)^+\|_F^2 \). By strong submultiplicativity,

\[
\|(A - A_k)S(V_k^T S)^+\|_F^2 \leq \|(A - A_k)S\|_F^2 \|(V_k^T S)^+\|_2^2.
\]

To conclude, we apply Lemma 13 to bound the two terms in the right-hand side of the above inequality. Again, the rescaling of the columns of \( C \) is irrelevant to the construction of \( \Pi_{C,k}^E(A) \). The running time of the proposed algorithm is equal to the time needed to compute \( V_k \) plus the time needed to compute \( A - A_k \) (which is equal to \( O(mnk) \) given \( V_k \)) plus the time needed to run the algorithm of Lemma 13, which is equal to \( O(nrk^2 + nm) \).

**Proof of Theorem 4.** We will follow the proof of Theorem 2, but, as with the proof of Theorem 3, instead of using the top \( k \) left singular vectors of \( A \) (the matrix \( V_k \)), we will use the matrix \( Z \) of Lemma 11 that is computed via a fast, approximate matrix factorization. More specifically, let \( Z \) be the matrix of Lemma 11 and run the algorithm of Lemma 13 on the following two sets of vectors: the \( n \) rows of the matrix \( Z \) and the \( n \) rows of the matrix \( E^T \). The output of the algorithm is a sampling and rescaling matrix \( S \in \mathbb{R}^{n \times r} \) (see discussion after Lemma 12 in Section 3.2). Let \( C = AS \) and note that \( C \) consists of a subset of \( r \) rescaled columns of \( A \). The proof of Theorem 4 is now identical to the proof of Theorem 2, except for using Lemma 8 instead of Lemma 9. Ultimately,
where in each trial the sampled columns and let $C$ be columns of $A$ (1 + $\epsilon$). Lemma 16. Given a matrix $A \in \mathbb{R}^{m \times n}$, a target rank $k$, and an integer $r$, there exists an algorithm to select $r$ columns from $A$ to form the matrix $C \in \mathbb{R}^{m \times r}$ such that

$$\mathbb{E} \left[ \|A - \Pi_{C,k}(A)\|_F^2 \right] \leq \|A - A_k\|_F^2 + \frac{k}{r} \|A\|_F^2.$$ 

The matrix $C$ can be computed in $O(mn + r \log r)$ time.

Algorithms for the above lemma choose $r$ columns of $A$ in $r$ independent identically distributed (i.i.d.) trials, where in each trial a column of $A$ is sampled with probability proportional to its norm-squared (importance sampling). We now state Theorem 2.1 of [9], which builds upon Lemma 15 to provide an adaptive sampling procedure that improves the accuracy guarantees of Lemma 15.

Lemma 16. Given a matrix $A \in \mathbb{R}^{m \times n}$, let $C_1 \in \mathbb{R}^{m \times r}$ consist of $r$ columns of $A$, and define the residual $B = A - C_1 C_1^T A \in \mathbb{R}^{m \times n}$. For $i = 1, \ldots, n$, let

$$p_i = \|b_i\|_2^2/\|B\|_F^2,$$

where $b_i$ is the $i$-th column of the matrix $B$. Sample a further $s$ columns from $A$ in $s$ i.i.d. trials, where in each trial the $i$-th column is chosen with probability $p_i$. Let $C_2 \in \mathbb{R}^{m \times s}$ contain the $s$ sampled columns and let $C = [C_1 \quad C_2] \in \mathbb{R}^{m \times (r+s)}$ contain the columns of both $C_1$ and $C_2$, all of which are columns of $A$. Then, for any integer $k > 0$,

$$\mathbb{E} \left[ \|A - \Pi_{C,k}(A)\|_F^2 \right] \leq \|A - A_k\|_F^2 + \frac{k}{s} \|B\|_F^2.$$ 

Note that Lemma 16 is an extension of Lemma 15, which can be derived by setting $C_1$ to be empty in Lemma 16. We are now ready to prove Theorem 5. First, fix $d > 1$ and define $c_0 = (1 + \epsilon_0) (1 + 1/(1 - \sqrt{k/r})^2)$, where $\hat{r} = \lceil dk \rceil$. (We will choose $d$ and $c_0$ later.) Now run the algorithm of Theorem 4 to sample $\hat{r} = \lceil dk \rceil$ columns of $A$ and form the matrix $C_1$. Then, run the adaptive sampling algorithm of Lemma 16 with $B = A - C_1 C_1^T A$ and sample a further $s = \lceil c_0 k/\epsilon \rceil$ columns of $A$ to form the matrix $C_2$. Let $C = [C_1 \quad C_2] \in \mathbb{R}^{m \times (r+s)}$ contain all the sampled columns. We will analyze the expectation $\mathbb{E} \left[ \|A - \Pi_{C,k}(A)\|_F^2 \right]$. Using the bound of Lemma 16, we first compute the expectation with respect to $C_2$ conditioned on $C_1$:

$$\mathbb{E}_{C_2} \left[ \|A - \Pi_{C,k}(A)\|_F^2 \right] \leq \|A - A_k\|_F^2 + \frac{k}{s} \|B\|_F^2.$$ 

The last inequality follows from the bounds of Lemma 13. The theorem now follows by taking the expectation of both sides and using Lemma 11 to bound $\mathbb{E} \left[ \|E\|_F^2 \right]$. Again, the rescaling of the columns of $C$ is irrelevant to the construction of $\Pi_{C,k}(A)$. The overall running time is derived by replacing the time needed to compute $V_k$ in Theorem 2 with the time needed to compute the fast approximate factorization of Lemma 11.

Proof of Theorem 5. Finally, we will prove Theorem 5 by combining the results of Theorem 4 (a constant factor approximation algorithm) with one round of adaptive sampling. We first recall the following lemma, which has appeared in prior work [10, 15].
We now compute the expectation with respect to $C_1$ (only $B$ depends on $C_1$):

$$\mathbb{E}_{C_1} \left[ \mathbb{E}_{C_2} \left[ \|A - \Pi_{C,k}^F(A)\|_F^2 | C_1 \right] \right] \leq \|A - A_k\|_F^2 + \frac{k}{s} \mathbb{E}_{C_1} \left[ \|A - C_1 C_1^T A\|_F^2 \right].$$

By the law of iterated expectation, the left hand side is exactly equal to $\mathbb{E} \left[ \|A - \Pi_{C,k}^F(A)\|_F^2 \right]$. We now use the accuracy guarantee of Theorem 4 and our definition of $c_0$ to bound

$$\mathbb{E}_{C_1} \left[ \|A - C_1 C_1^T A\|_F^2 \right] \leq \mathbb{E}_{C_1} \left[ \|A - \Pi_{C_1,k}^F(A)\|_F^2 \right] \leq c_0\|A - A_k\|_F^2.$$

Using the bound in (4), we obtain

$$\mathbb{E} \left[ \|A - \Pi_{C,k}^F(A)\|_F^2 \right] \leq \|A - A_k\|_F^2 \left(1 + c_0k/s \right).$$

Finally, recall that for our choice of $s$, $s \geq c_0k/\epsilon$, and so we obtain the relative error bound. The number of columns needed is $r = \hat{r} + s = dk + c_0k/\epsilon$. Set $d = (1 + \alpha)^2$, where $\alpha = \sqrt{(1 + c_0)/\epsilon}$. After some algebra, this yields $r = k(\alpha^3 + (1 + \alpha)^3) = \frac{2k}{\epsilon} (1 + O(\epsilon_0 + \epsilon^{1/3}))$ sampled columns. The time needed to compute the matrix $C$ is the sum of three terms: the running time of Theorem 4 (which is $O(mnk_0 + nrk^2)$), plus the time needed to compute $A - C_1 C_1^T A$ (which is $O(mnr)$), plus the time needed to run the algorithm of Lemma 16 (which is $O(mn + s \log s)$). Assume $r < n$ (otherwise the problem is trivial), set $\epsilon_0 = \epsilon^{2/3}$ and use $d = O(\epsilon^{-2/3})$ to get the final asymptotic run time.

We conclude by noting that the number of columns required for relative error approximation is approximately $\frac{2k}{\epsilon}$, a two-factor from optimal, since $\frac{k}{\epsilon}$ columns are necessary (see [8] and Section 9.1). We get an improved running time equal to $O(mnk + nk^3 + n \log \epsilon^{-1})$ using just a constant factor more columns by setting $d$ and $\epsilon_0$ in the proof to constants (for example, setting $d = 100$, $\epsilon_0 = \frac{62}{181} \approx \frac{1}{3}$ results in sampling $\frac{2k}{\epsilon} (1 + o(1))$ columns).

## 5 Proof of Lemma 10: Approximate SVD in the Spectral Norm

Consider the following algorithm, described in Corollary 10.10 of [19]. The algorithm takes as inputs a matrix $A \in \mathbb{R}^{m \times n}$ of rank $\rho$, an integer $2 \leq k < \rho$, an integer $q \geq 1$, and an integer $p \geq 2$. Set $r = k + p$ and construct the matrix $Y \in \mathbb{R}^{m \times r}$ as follows:

1. Generate an $n \times r$ standard Gaussian matrix $R$ whose entries are i.i.d. $\mathcal{N}(0,1)$ variables.
2. Return $Y = (AA^T)^q A R \in \mathbb{R}^{m \times r}$.

The running time of the above algorithm is $O(mnrq)$. Corollary 10.10 of [19] presents the following bound:

$$\mathbb{E} \left[ \|A - YY^+ A\|_2 \right] \leq \left(1 + \sqrt{\frac{k}{p - 1}} + \frac{e\sqrt{k + p}}{p} \sqrt{\min\{m,n\} - k} \right)^{\frac{1}{2+p}} \|A - A_k\|_2,$$

where $e = 2.718 \ldots$. To the best of our understanding, the above result is not immediately applicable to the construction of a factorization of the form $A = BZ^T + E$, because $Y$ contains $r > k$ columns. Lemma 25 below, which strengthens Corollary 10.10 in [19], argues that the matrix $\Pi_{Y,k}^2(A)$ “contains” the desired factorization $BZ^T$. Recall that while we cannot compute $\Pi_{Y,k}^2(A)$ efficiently, we can compute a constant-factor approximation, which is sufficient for our purposes. The proof of Lemma 25 is very similar to the proof of Corollary 10.10 of [19], with the only difference being our starting point: instead of using Theorem 9.1 of [19] we use Lemma 9 of our work. To prove Lemma 25, we will need several results for standard Gaussian matrices, projection matrices, and Hölder’s inequality. The following seven lemmas are all borrowed from [19].
Lemma 17 (Proposition 10.1 in [19]). Fix matrices $X$, $Y$, and draw a standard Gaussian matrix $R$ of appropriate dimensions. Then,

$$
\mathbb{E} \left[ \|XRY\|_2 \right] \leq \|X\|_2 \|Y\|_F + \|X\|_F \|Y\|_2.
$$

Lemma 18 (Proposition 10.2 in [19]). For $k, p \geq 2$, draw a standard Gaussian matrix $R \in \mathbb{R}^{k \times (k+p)}$. Then,

$$
\mathbb{E} \left[ \|R^+\|_2 \right] \leq \frac{e \sqrt{k + p}}{p},
$$

where $e = 2.718 \ldots$.

Lemma 19 (Proposition 10.1 in [19]). Fix matrices $X$, $Y$, and a standard Gaussian matrix $R$ of appropriate dimensions. Then,

$$
\mathbb{E} \left[ \|XRY\|_F^2 \right] = \|X\|_F^2 \|Y\|_F^2.
$$

Lemma 20 (Proposition 10.2 in [19]). For $k, p \geq 2$, draw a standard Gaussian matrix $R \in \mathbb{R}^{k \times (k+p)}$. Then,

$$
\mathbb{E} \left[ \|R^+\|_F^2 \right] = \frac{k}{p - 1}.
$$

Lemma 21 (proved in [19]). For integers $k, p \geq 1$, and a standard Gaussian $R \in \mathbb{R}^{k \times (k+p)}$ the rank of $R$ is equal to $k$ with probability one.

Lemma 22 (Proposition 8.6 in [19]). Let $P$ be a projection matrix. For any matrix $X$ of appropriate dimensions and an integer $q \geq 0$,

$$
\|PX\|_2 \leq \left( \|P(XX^T)^qX\|_2 \right)^{\frac{1}{2q+1}}.
$$

Lemma 23 (Hölder’s inequality). Let $x$ be a positive random variable. Then, for any $h \geq 1$,

$$
\mathbb{E} [x] \leq \left( \mathbb{E} \left[ x^h \right] \right)^{\frac{1}{h}}.
$$

The following lemma provides an alternative definition for $\Pi_{C,k}^\xi(A)$ which will be useful in subsequent proofs. Recall from Section 2.2 that we can write $\Pi_{C,k}^\xi(A) = CX^\xi$, where

$$
X^\xi = \arg\min_{\Psi \in \mathbb{R}^{r \times n}: \text{rank}(\Psi) \leq k} \|A - C\Psi\|_2^2.
$$

The next lemma basically says that $\Pi_{C,k}^\xi(A)$ is the projection of $A$ onto the rank-$k$ subspace spanned by $CX^\xi$, and that no other subspace in the column space of $C$ is better.

Lemma 24. For $A \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{m \times r}$, integer $r > k$, let $\Pi_{C,k}^\xi(A) = CX^\xi$, and $Y \in \mathbb{R}^{r \times n}$ be any matrix of rank at most $k$. Then,

$$
\|A - CX^\xi\|_2^2 = \|A - (CX^\xi)(CX^\xi)^+A\|_2^2 \leq \|A - (CY)(CY)^+A\|_2^2,
$$

where $Y \in \mathbb{R}^{r \times n}$ is any matrix of rank at most $k$.

Proof. The second inequality will follow from the optimality of $X^\xi$ because $Y(CY)^+A$ has rank at most $k$. So we only need to prove the first equality. Again, by the optimality of $X^\xi$ and because $X^\xi(CX^\xi)^+A$ has rank at most $k$, $\|A - CX^\xi\|_2^2 \leq \|A - (CX^\xi)(CX^\xi)^+A\|_2^2$. To get the reverse inequality, we will use matrix-Pythagoras as follows:

$$
\|A - CX^\xi\|_2^2 = \| \begin{pmatrix} I_m - (CX^\xi)(CX^\xi)^+ \end{pmatrix} A - CX^\xi(I_n - (CX^\xi)^+A) \|_2^2 \\
\geq \| \begin{pmatrix} I_m - (CX^\xi)(CX^\xi)^+ \end{pmatrix} A \|_2^2.
$$
Lemma 25 (Extension of Corollary 10.10 of [19]). Let \( A \) be a matrix in \( \mathbb{R}^{m \times n} \) of rank \( \rho \), let \( k \) be an integer satisfying \( 2 \leq k < \rho \), and let \( r = k + p \) for some integer \( p \geq 2 \). Let \( R \in \mathbb{R}^{n \times r} \) be a standard Gaussian matrix (i.e., a matrix whose entries are drawn in i.i.d. trials from \( \mathcal{N}(0,1) \)). Define \( B = (AA^T)^qA \) and compute \( Y = BR \). Then, for any \( q \geq 0 \),

\[
\mathbb{E} \left[ \|A - \Pi_{Y,k}^2(A)\|_2 \right] \leq \left( 1 + \sqrt{\frac{k}{p-1}} + \frac{e\sqrt{k + p}}{p} \sqrt{\min\{m,n\} - k} \right)^{\frac{1}{2q+1}} \|A - A_k\|_2
\]

Proof. Let \( \Pi_{Y,k}^2(A) = YX_1 \) and \( \Pi_{Y,k}^2(B) = YX_2 \), where \( X_1 \) is optimal for \( A \) and \( X_2 \) for \( B \). From Lemma 24,

\[
\|A - \Pi_{Y,k}^2(A)\|_2 = \| (I_m - (YX_1)(YX_1)^+)A\|_2 \leq \| (I_m - (YX_2)(YX_2)^+)A\|_2.
\]

From Lemma 22 and using the fact that \( I_m - (YX_2)(YX_2)^+ \) is a projection,

\[
\| (I_m - (YX_2)(YX_2)^+)A\|_2 \leq \| B - (YX_2)(YX_2)^+B\|_2^{\frac{1}{2q+1}}
\]

where the last step follows from Lemma 24. We conclude that

\[
\|A - \Pi_{Y,k}^2(A)\|_2 \leq \| B - \Pi_{Y,k}^2(B)\|_2^{\frac{1}{2q+1}}.
\]

\( Y \) is generated using a random \( R \), so taking expectations and applying Hölder’s inequality, we get

\[
\mathbb{E} \left[ \|A - \Pi_{Y,k}^2(A)\|_2 \right] \leq \left( \mathbb{E} \left[ \| B - \Pi_{Y,k}^2(B)\|_2 \right] \right)^{\frac{1}{2q+1}}. \tag{1}
\]

We now focus on bounding the term on the right-hand side of the above equation. Let the SVD of \( B \) be \( B = U_B \Sigma_B V_B^T \), with the top rank \( k \) factors from the SVD of \( B \) being \( U_{B,k}, \Sigma_{B,k} \), and \( V_{B,k} \) and the corresponding trailing factors being \( U_{B,r}, \Sigma_{B,r} \) and \( V_{B,r} \). Let \( \rho_B \) be the rank of \( B \). Let

\[
\Omega_1 = V_{B,k}^T R \in \mathbb{R}^{k \times r} \quad \text{and} \quad \Omega_2 = V_{B,r}^T R \in \mathbb{R}^{(\rho_B-k) \times r}.
\]

The Gaussian distribution is rotationally invariant, so \( \Omega_1, \Omega_2 \) are also standard Gaussian matrices which are stochastically independent because \( V_B^T \) can be extended to a full rotation. Thus, \( V_{B,k}^T R \) and \( V_{B,r}^T R \) also have entries that are i.i.d. \( \mathcal{N}(0,1) \) variables. We now apply Lemma 9 to reconstruct \( B \), with \( \xi = 2 \) and \( S = R \). The rank requirement in Lemma 9 is satisfied because, from Lemma 21, the rank of \( \Omega_1 \) is equal to \( k \) (as it is a standard normal matrix), and thus the matrix \( R \) satisfies the rank assumptions of Lemma 9. We get that,

\[
\|B - \Pi_{Y,k}^2(B)\|_2^2 \leq \|B - B_k\|_2^2 + \| (B - B_k)R(V_{B,k}^T R)^+\|_2^2 \leq \| \Sigma_{B,r} \|_2^2 + \| \Sigma_{B,r} \Omega_2 \Omega_2^+ \|_2^2.
\]

Using \( \sqrt{x^2 + y^2} \leq x + y \), we conclude that \( \|B - \Pi_{Y,k}^2(B)\|_2 \leq \| \Sigma_{B,r} \|_2 + \| \Sigma_{B,r} \Omega_2 \Omega_2^+ \|_2 \). We now need to take the expectation with respect to \( \Omega_1, \Omega_2 \). We first take the expectation with respect to \( \Omega_2 \), conditioning on \( \Omega_1 \). We then take the expectation w.r.t. \( \Omega_1 \). Since only the second term is stochastic, using Lemma 17, we have:

\[
\mathbb{E}_{\Omega_2} \left[ \| \Sigma_{B,r} \Omega_2 \Omega_2^+ \|_2 \right] \leq \| \Sigma_{B,r} \|_2 \| \Omega_2^+ \|_F + \| \Sigma_{B,r} \|_F \| \Omega_1^+ \|_2
\]

We now take the expectation with respect to \( \Omega_1 \). To bound \( \mathbb{E} \left[ \| \Omega_1^+ \|_2 \right] \), we use Lemma 18. To bound \( \mathbb{E} \left[ \| \Omega_1^+ \|_F \right] \), we first use Hölder’s inequality to bound \( \mathbb{E} \left[ \| \Omega_1^+ \|_F \right] \leq \mathbb{E} \left[ \| \Omega_1^+ \|_2 \right]^{1/2} \), and then
we use Lemma 20. Since \( \| \Sigma_{B, \tau} \|_F \leq \sqrt{\min(m, n) - k} \| \Sigma_{B, \tau} \|_2 \), collecting our results together, we obtain:

\[
\mathbb{E} [ \| B - \Pi_{Y, k}^2(B) \|_2 ] \leq \left( 1 + \sqrt{\frac{k}{p - 1} + \frac{e\sqrt{k + p}}{p} \sqrt{\min(m, n) - k}} \right) \| B - B_k \|_2.
\]

To conclude, combine with eqn. (1) and note that \( \| B - B_k \|_2 = \| A - A_k \|^{2q+1} \).

We now have all the necessary ingredients to prove Lemma 10. Let \( Y \) be the matrix of Lemma 25. Set \( p = k \) and

\[
q = \left\lceil \log \left( 1 + \sqrt{\frac{k}{p - 1} + \frac{e\sqrt{k + p}}{p} \sqrt{\min(m, n) - k}} \right) \right\rceil,
\]

so that

\[
\left( 1 + \sqrt{\frac{k}{p - 1} + \frac{e\sqrt{k + p}}{p} \sqrt{\min(m, n) - k}} \right)^{2q+1} \leq 1 + \frac{\epsilon}{\sqrt{2}}.
\]

Then,

\[
\mathbb{E} [ \| A - \Pi_{Y, k}^2(A) \|_2 ] \leq \left( 1 + \frac{\epsilon}{\sqrt{2}} \right) \| A - A_k \|_2. \tag{2}
\]

Given \( Y \), let \( Q \) be an orthonormal basis for its column space. Then, using the algorithm of Section 2.2 and applying Lemma 7 we can construct the matrix \( Q (Q^T A)_k \) such that

\[
\| A - Q (Q^T A)_k \|_2 \leq \sqrt{2} \| A - \Pi_{Y, k}^2(A) \|_2.
\]

Clearly, \( Q (Q^T A)_k \) is a rank \( k \) matrix; let \( Z \in \mathbb{R}^{n \times k} \) denote the matrix containing the right singular vectors of \( Q (Q^T A)_k \), so \( Q (Q^T A)_k = XZ^T \). Note that \( Z \) is equal to the right singular vectors of the matrix \( (Q^T A)_k \) (because \( Q \) has orthonormal columns), and so \( Z \) has already been computed at the second step of the algorithm of Section 2.2. Since \( E = A - AZZ^T \) and \( \| A - AZZ^T \|_2 \leq \| A - XZ^T \|_2 \) for any \( X \), we have

\[
\| E \|_2 \leq \| A - Q (Q^T A)_k \|_2 \leq \sqrt{2} \| A - \Pi_{Y, k}^2(A) \|_2.
\]

Note that, by construction, \( EZ = 0_{m \times k} \). The running time follows by adding the running time of the algorithm at the beginning of this section and the running time of the algorithm of Lemma 2.2.

6 Proof of Lemma 11: Approximate SVD in the Frobenius Norm

Consider the following algorithm, described in Theorem 10.5 of [19]. The algorithm takes as inputs a matrix \( A \in \mathbb{R}^{m \times n} \) of rank \( \rho \), an integer \( 2 \leq k < \rho \), and an integer \( p \geq 2 \). Set \( r = k + p \) and construct the matrix \( Y \in \mathbb{R}^{m \times r} \) as follows:

1. Generate an \( n \times r \) standard Gaussian matrix \( R \) whose entries are i.i.d. \( N(0,1) \) variables.
2. Return \( Y = AR \in \mathbb{R}^{m \times r} \).

The running time of the above algorithm is \( O(mnr) \). Theorem 10.5 in [19] presents the following bound:

\[
\mathbb{E} [ \| A - YY^+ A \|_F ] \leq \left( 1 + \frac{k}{p - 1} \right) \frac{1}{2} \| A - A_k \|_F.
\]

To the best of our understanding, the above result is not immediately applicable to the construction of a factorization of the form \( A = BZ^T + E \) (as in Lemma 11) because \( Y \) contains \( r > k \) columns.
Lemma 26 (Extension of Theorem 10.5 of [19]). Let $A$ be a matrix in $\mathbb{R}^{m \times n}$ of rank $\rho$, let $k$ be an integer satisfying $2 \leq k < \rho$, and let $r = k + p$ for some integer $p \geq 2$. Let $R \in \mathbb{R}^{n \times r}$ be a standard Gaussian matrix (i.e., a matrix whose entries are drawn in i.i.d. trials from $\mathcal{N}(0,1)$) and compute $Y = AR$. Then,

$$
\mathbb{E} \left[ \|A - \Pi_{Y,k}(A)\|_F^2 \right] \leq \left( 1 + \frac{k}{p-1} \right) \|A - A_k\|_F^2.
$$

**Proof.** We construct the matrix $Y$ as described in the beginning of this section. Let the rank of $A$ be $\rho$ and let $A = U \Sigma V^T$ be the SVD of $A$. Define

$$
\Omega_1 = V_k^T R \in \mathbb{R}^{k \times r} \quad \text{and} \quad \Omega_2 = V_{\rho-k}^T R \in \mathbb{R}^{(\rho-k) \times r}.
$$

The Gaussian distribution is rotationally invariant, so $\Omega_1$, $\Omega_2$ are also standard Gaussian matrices which are stochastically independent because $V^T$ can be extended to a full rotation. Thus, $V_k^T R$ and $V_{\rho-k}^T R$ also have entries that are i.i.d. $\mathcal{N}(0,1)$ variables. We now apply Lemma 9 to reconstructing $A$, with $\xi = F$ and $S = R$. Recall that from Lemma 21, the rank of $\Omega_1$ is equal to $k$, and thus the matrix $R$ satisfies the rank assumptions of Lemma 9. We have that,

$$
\|A - \Pi_{Y,k}(A)\|_F^2 \leq \|A - A_k\|_F^2 + \|\Sigma_{\rho-k} \Omega_2 \Omega_1^+\|_F^2,
$$

where $A - A_k = U_{\rho-k} \Sigma_{\rho-k} V_{\rho-k}^T$. To conclude, we take the expectation on both sides, and since only the second term on the right hand side is stochastic, we bound as follows:

$$
\mathbb{E} \left[ \|\Sigma_{\rho-k} \Omega_2 \Omega_1^+\|_F^2 \right] \overset{(a)}{=} \mathbb{E}_{\Omega_1} \left[ \mathbb{E}_{\Omega_2} \left[ \|\Sigma_{\rho-k} \Omega_2 \Omega_1^+\|_F^2 | \Omega_1 \right] \right]
$$

$$
\overset{(b)}{=} \mathbb{E}_{\Omega_1} \left[ \|\Sigma_{\rho-k}\|_F^2 \|\Omega_1^+\|_F^2 \right]
$$

$$
\overset{(c)}{=} \|\Sigma_{\rho-k}\|_F^2 \mathbb{E} \left[ \|\Omega_1^+\|_F^2 \right]
$$

$$
\overset{(d)}{=} \frac{k}{p-1} \|\Sigma_{\rho-k}\|_F^2.
$$

(a) follows from the law of iterated expectation; (b) follows from Lemma 19; (c) follows because $\|\Sigma_{\rho-k}\|_F^2$ is a constant; (d) follows from Lemma 20. We conclude the proof by noting that $\|\Sigma_{\rho-k}\|_F = \|A - A_k\|_F$. 

We now have all the necessary ingredients to conclude the proof of Lemma 11. Let $Y$ be the matrix of Lemma 26, and let $Q$ be an orthonormal basis for its column space. Then, using the algorithm of Section 2.2 and applying Lemma 7 we can construct the matrix $Q (Q^T A)_k$ such that

$$
\|A - Q (Q^T A)_k\|_F^2 = \|A - \Pi_{Y,k}(A)\|_F^2.
$$

Clearly, $Q (Q^T A)_k$ is a rank $k$ matrix; let $Z \in \mathbb{R}^{n \times k}$ be the matrix containing the right singular vectors of $Q (Q^T A)_k$, so $Q (Q^T A)_k = XZ^T$. Note that $Z$ is equal to the right singular vectors of the matrix $(Q^T A)_k$ (because $Q$ has orthonormal columns), and thus $Z$ has already been computed at the second step of the algorithm of Section 2.2. Since $E = A - AZZ^T$ and $\|A - AZZ^T\|_F \leq \|A - XZ^T\|_F$ for any $X$, we have

$$
\|E\|_F^2 \leq \|A - Q (Q^T A)_k\|_F^2 = \|A - \Pi_{Y,k}(A)\|_F^2.
$$

To conclude, take expectations on both sides, use Lemma 26 to bound $\mathbb{E} \left[ \|A - \Pi_{Y,k}(A)\|_F^2 \right]$, and set $p = \lceil \frac{k}{\epsilon} + 1 \rceil$ to obtain:

$$
\mathbb{E} \left[ \|E\|_F^2 \right] \leq \left( 1 + \frac{k}{p-1} \right) \|A - A_k\|_F^2 \leq (1 + \epsilon) \|A - A_k\|_F^2,
$$

By construction, $EZ = 0_{m \times k}$. The running time follows by adding the running time of the algorithm at the beginning of this section and the running time of the algorithm of Lemma 2.2.
Input:

- $V = \{v_1, \ldots, v_n\}$, with $\sum_{i=1}^{n} v_i v_i^T = I_k$ ($k < n$)
- $U = \{u_1, \ldots, u_n\}$, with $\sum_{i=1}^{n} u_i u_i^T = I_\ell$ ($\ell \leq n$)
- integer $r$, with $k < r < n$

Output: A vector of weights $s = [s_1, \ldots, s_n]$, with $s_i \geq 0$ and at most $r$ non-zero $s_i$'s.

1. Initialize $s_0 = 0_{n \times 1}$, $A_0 = 0_{k \times k}$, $B_0 = 0_{\ell \times \ell}$.

2. For $\tau = 0, \ldots, r - 1$
   - Compute $l_\tau$ and $u_\tau$ from eqn. (6).
   - Find an index $j$ in $\{1, \ldots, n\}$ such that
     \[ U(u_j, \delta_u, B_\tau, u_\tau) \leq L(v_j, \delta_l, A_\tau, l_\tau). \]
   - Let
     \[ t^{-1} = \frac{1}{2} (U(u_j, \delta_u, B_\tau, u_\tau) + L(v_j, \delta_l, A_\tau, l_\tau)). \]
     
   - Update the $j$th component of $s$, $A_\tau$ and $B_\tau$:
     \[ s_{\tau+1}[j] = s_\tau[j] + t, \ A_{\tau+1} = A_\tau + tv_j v_j^T, \text{ and } B_{\tau+1} = B_\tau + tu_j u_j^T. \]

3. Return $s = r^{-1} \left(1 - \sqrt{k/r}\right) \cdot s_r$.

**Algorithm 1:** Deterministic Dual Set Spectral Sparsification.

### 7 Dual Set Spectral Sparsification: proof of Lemma 12

In this section, we prove Lemma 12, which generalizes Theorem 3.1 in [1]. Indeed, setting $V = U$ reproduces the spectral sparsification result of Theorem 3.1 in [1]. We will provide a constructive proof of the lemma and we start by describing the algorithm that computes the weights $s_i$, $i = 1, \ldots, n$.

#### 7.1 The Algorithm

The fundamental idea underlying Algorithm 1 is the greedy selection of vectors that satisfy a number of desired properties in each step. These properties will eventually imply the eigenvalue bounds of Lemma 12. We start by defining several quantities that will be used in the description of the algorithm and its proof. First, fix two constants:

\[ \delta_l = 1; \quad \delta_u = \frac{1 + \sqrt{\frac{\ell}{r}}}{1 - \sqrt{\frac{\ell}{r}}}. \]
Given \( k, \ell, \) and \( r \) (all inputs of Algorithm 1), and a parameter \( \tau = 0, \ldots, r-1 \), define two parameters \( L_\tau \) and \( U_\tau \) as follows:

\[
L_\tau = r \left( \frac{\tau}{r} - \sqrt{\frac{k}{r}} \right) = \tau - \sqrt{rk}; \quad U_\tau = \frac{\left( \tau - r \right) \left( 1 + \sqrt{\frac{\tau}{r}} \right) + r \left( 1 + \sqrt{\frac{\tau}{r}} \right)^2}{1 - \sqrt{\frac{k}{r}}} = \delta_u \left( \tau + \sqrt{\ell r} \right). \tag{6}
\]

We next define the lower and upper functions \( \phi(l, A) \) (\( l \in \mathbb{R} \) and \( A \in \mathbb{R}^{k \times k} \)) and \( \varphi(u, B) \) (\( u \in \mathbb{R} \) and \( B \in \mathbb{R}^{\ell \times \ell} \)) as follows:

\[
\phi(l, A) = \sum_{i=1}^{k} \frac{1}{\lambda_i(A) - l}; \quad \varphi(u, B) = \sum_{i=1}^{\ell} \frac{1}{u - \lambda_i(B)}. \tag{7}
\]

Let \( L(v, \delta_l, A, l) \) be a function with four inputs (a vector \( v \in \mathbb{R}^{k \times 1} \), \( \delta_l \in \mathbb{R} \), a matrix \( A \in \mathbb{R}^{k \times k} \), and \( l \in \mathbb{R} \)):

\[
L(v, \delta_l, A, l) = \frac{v^T(A - (l + \delta_l)I_k)^{-2}v}{\phi(l + \delta_l, A) - \phi(l, A)} - v^T(A - (l + \delta_l)I_k)^{-1}v. \tag{8}
\]

Similarly, let \( U(u, \delta_u, B, u) \) be a function with four inputs (a vector \( u \in \mathbb{R}^{\ell \times 1} \), \( \delta_u \in \mathbb{R} \), a matrix \( B \in \mathbb{R}^{\ell \times \ell} \), and \( u \in \mathbb{R} \)):

\[
U(u, \delta_u, B, u) = \frac{u^T((u + \delta_u)I_\ell - B)^{-2}u}{\phi(u, B) - \phi(u + \delta_u, B)} + u^T((u + \delta_u)I_\ell - B)^{-1}u. \tag{9}
\]

Algorithm 1 runs in \( r \) steps. The initial vector of weights \( s_0 \) is initialized to the all-zero vector. At each step \( \tau = 0, \ldots, r-1 \), the algorithm selects a pair of vectors \((u_j, v_j)\) that satisfy eqn. (3), computes the associated weight \( t \) from eqn. (4), and updates two matrices and the vector of weights appropriately, as specified in eqn. (5).

### 7.2 Running time

The algorithm runs in \( r \) iterations. In each iteration, we evaluate the functions \( U(u, \delta_u, B, u) \) and \( L(v, \delta_l, A, l) \) at most \( n \) times. Note that all \( n \) evaluations for both functions need at most \( O(k^3 + nk^2 + \ell^3 + n\ell^2) \) time, because the matrix inversions can be performed once for all \( n \) evaluations. Finally, the updating step needs an additional \( O(k^2 + \ell^2) \) time. Overall, the complexity of the algorithm is of the order \( O(r(k^3 + nk^2 + \ell^3 + n\ell^2 + k^2 + \ell^2)) = O(rn(k^2 + \ell^2)) \).

Note that when \( \mathcal{U} \) is the standard basis \( \{\mathcal{U} = \{e_1, \ldots, e_n\} \) and \( \ell = n \), the computations can be done much more efficiently: the eigenvalues of \( B_s \) need not be computed explicitly (the expensive step), since they are available by inspection, being equal to the weights \( s_\tau \). In the function \( U(u, \delta_u, B, u) \), the functions \( \phi \) (given the eigenvalues) need only be computed once per iteration, in \( O(n) \) time. The remaining terms can be computed in \( O(1) \) time, because, for example, \( e_i^T((u + \delta_u)I - B)^{-2}e_i = (u + \delta_u - s_i e_i)^{-2} \). The running time now drops to \( O(rnk^2) \), since all the operations on \( \mathcal{U} \) only contribute \( O(rn) \).

### 7.3 Proof of Correctness

We prove that the output of Algorithm 1 satisfies Lemma 12. Our proof is similar to the proof of Theorem 3.1 [1]. The main difference is that we need to accommodate two different sets of vectors. Let \( W \in \mathbb{R}^{m \times m} \) be a positive semi-definite matrix with eigendecomposition

\[
W = \sum_{i=1}^{m} \lambda_i(W) u_iu_i^T
\]
and recall the functions $\bar{\phi}(l, W)$, $\bar{\phi}(u, W)$, $L(v, \delta_l, W, l)$, and $U(v, \delta_u, W, u)$ defined in eqns. (7), (8), and (9). We now quote two lemmas proven in [1] using the Sherman-Morrison-Woodbury identity; these lemmas allow one to control the smallest and largest eigenvalues of $W$ under a rank-one perturbation.

**Lemma 27.** Fix $\delta_l > 0$, $W \in \mathbb{R}^{m \times m}$, $v \in \mathbb{R}^m$, and $L < \lambda_m(W)$. If $t > 0$ satisfies
\[
t^{-1} \leq L(v, \delta_l, W, l),
\]
then $\lambda_m(W + tvv^T) \geq L + \delta_l$.

**Lemma 28.** Fix $\delta_u > 0$, $W \in \mathbb{R}^{m \times m}$, $v \in \mathbb{R}^m$, and $v > \lambda_1(W)$. If $t$ satisfies
\[
t^{-1} \geq U(v, \delta_u, W, u),
\]
then $\lambda_1(W + tvv^T) \leq v + \delta_u$.

Now recall that Algorithm 1 runs in $r$ steps. Initially, all $n$ weights are set to zero. Assume that at the $\tau$-th step ($\tau = 0, \ldots, r - 1$) the vector of weights $s_\tau = [s_\tau[1], \ldots, s_\tau[n]]$ has been constructed and let
\[
A_\tau = \sum_{i=1}^{n} s_\tau[i]v_i v_i^T \quad \text{and} \quad B_\tau = \sum_{i=1}^{n} s_\tau[i]u_i u_i^T.
\]
Note that both matrices $A_\tau$ and $B_\tau$ are positive semi-definite. We claim the following lemma which guarantees that the algorithm is well-defined. The proof is deferred to the next subsection.

**Lemma 29.** At the $\tau$-th step, for all $\tau = 0, \ldots, r - 1$, there exists an index $j$ in $\{1, \ldots, n\}$ such that setting the weight $t > 0$ as in eqn. (4) satisfies
\[
U(u_j, \delta_u, B_\tau, u_\tau) \leq t^{-1} \leq L(v_j, \delta_l, A_\tau, l_\tau).
\]

Once an index $j$ and a weight $t > 0$ have been computed, Algorithm 1 updates the $j$-th weight in the vector of weights $s_\tau$ to create the vector of weights $s_{\tau+1}$. Clearly, at each of the $r$ steps, only one element of the vector of weights is updated. Since $s_0$ is initialized to the all-zeros vector, after all $r$ steps are completed, at most $r$ weights are non-zero. The following lemma argues that $\lambda_{\min}(A_\tau)$ and $\lambda_{\max}(B_\tau)$ are bounded.

**Lemma 30.** At the $\tau$-th step, for all $\tau = 0, \ldots, r - 1$, $\lambda_{\min}(A_\tau) \geq L_\tau$ and $\lambda_{\max}(B_\tau) \leq U_\tau$.

**Proof.** Recall eqn. (6) and observe that $L_0 = -\sqrt{r}k < 0$ and $U_0 = \delta_u \sqrt{r}l > 0$. Thus, the lemma holds at $\tau = 0$. It is also easy to verify that $L_{\tau+1} = L_\tau + \delta_l$, and, similarly, $U_{\tau+1} = U_\tau + \delta_u$. Now, at the $\tau$-step, given an index $j$ and a corresponding weight $t > 0$ satisfying eqn. (10), Lemmas 27 and 28 imply that
\[
\lambda_{\min}(A_{\tau+1}) = \lambda_{\min}(A_\tau + tv_j v_j^T) \geq L_\tau + \delta_l = L_{\tau+1};
\]
\[
\lambda_{\min}(B_{\tau+1}) = \lambda_{\max}(B_\tau + tu_j u_j^T) \leq U_\tau + \delta_u = U_{\tau+1}.
\]
The lemma now follows by simple induction on $\tau$.

We are now ready to conclude the proof of Lemma 12. By Lemma 30, at the $r$-th step,
\[
\lambda_{\max}(B_r) \leq U_r \quad \text{and} \quad \lambda_{\min}(A_r) \geq L_r.
\]
Recall the definitions of $U_r$ and $L_r$ from eqn. (6) and note that they are both positive and well-defined because $r > k$. Lemma 12 now follows after rescaling the vector of weights $s$ by $r^{-1} \left(1 - \sqrt{k/r}\right)$. Note that the rescaling does not change the number of non-zero elements of $s$, but does rescale all the eigenvalues of $A_r$ and $B_r$. 

20
7.4 Proof of Lemma 29

In order to prove Lemma 29 we will use the following averaging argument.

Lemma 31. At any step \( \tau = 0, \ldots, r - 1, \)

\[
\sum_{i=1}^{n} U(u_i, \delta_i, B, u) \leq 1 - \sqrt{\frac{k}{r}} \leq \sum_{i=1}^{n} L(v_i, \delta_i, A, L).
\]

Proof. For notational convenience, let \( \bar{\phi} = \bar{\phi}(u, B) \) and \( \bar{\phi} = \bar{\phi}(l, A) \). At \( \tau = 0 \), \( B_0 = 0 \) and \( A_0 = 0 \) and thus \( \bar{\phi} = \phi(u, B) \) and \( \bar{\phi} = \phi(l, A) \). For notational convenience, let \( \delta_0 = \delta / \sqrt{r} \) and \( \phi_0 = \phi_0 = -k / \ell_0 \). Focus on the \( \tau \)-th step and assume that the algorithm has run correctly up to that point. Then, \( \bar{\phi} \leq \bar{\phi} \) and \( \bar{\phi} \leq \phi_0 \). Both are true at \( \tau = 0 \) and, assuming that the algorithm has run correctly until the \( \tau \)-th step, Lemmas 27 and 28 guarantee that \( \bar{\phi} \) and \( \bar{\phi} \) are non-increasing.

First, consider the upper bound on \( U \). In the following derivation, \( \lambda_i \) denotes the \( i \)-th eigenvalue of \( B_T \). Using \( \text{Tr}(u^T Xu) = \text{Tr}(Xuu^T) \) and \( \sum_i u_i u_i^T = I \), we get

\[
\sum_{i=1}^{n} U(u_i, \delta_i, B, u, \tau) = \frac{\text{Tr}[(u_{\tau+1}^T I - B_T)^2]}{\Delta \bar{\phi} - \bar{\phi}(u_{\tau+1}, B)} + \bar{\phi}(u_{\tau+1}, B)
\]

\[
= \frac{\sum_{i=1}^{\ell} \frac{1}{(u_{\tau+1}^T - \lambda_i)^2}}{\Delta \bar{\phi}} + \frac{1}{\Delta \bar{\phi}} \sum_{i=1}^{\ell} \frac{1}{(u_{\tau+1}^T - \lambda_i)}
\]

\[
= \left( 1 - \frac{1}{\Delta \bar{\phi}} \sum_{i=1}^{\ell} \frac{1}{(u_{\tau} - \lambda_i)^2} \right) - \delta_0 \sum_{i=1}^{\ell} \left( u_{\tau} - \lambda_i \right) - \delta_0 \bar{\phi}_0.
\]

The last line follows because the last two terms are negative (using the fact that \( u_{\tau+1} > u_{\tau} > \lambda_i \) and \( \bar{\phi} \leq \phi_0 \)). Now, using \( \phi_0 = \delta \sqrt{r} \) and the definition of \( \delta \), the upper bound follows:

\[
\frac{1}{\delta} + \phi_0 = \frac{1}{\delta} + \frac{\ell}{\delta \sqrt{r}} = \frac{1}{\delta} \left( 1 + \frac{\ell}{\tau} \right) = 1 - \sqrt{\frac{k}{r}}.
\]

In order to prove the lower bound on \( L \) we use a similar argument. Let \( \lambda_i \) denote the \( i \)-th eigenvalue of \( A_T \). Then,

\[
\sum_{i=1}^{n} L(v_i, \delta_i, A, L) = \frac{\text{Tr}[\left(A_T - L_T + I_k\right)^2]}{\Delta \bar{\phi}(L_T + 1, A_T)} - \bar{\phi}(L_T + 1, A_T)
\]

\[
= \sum_{i=1}^{k} \frac{1}{\lambda_i^2 - (L_T + 1)(\lambda_i - L_T)} - \sum_{i=1}^{k} \frac{1}{\lambda_i - L_T + 1)
\]

\[
= \left( \frac{\sum_{i=1}^{k} \frac{1}{(\lambda_i - L_T + 1)(\lambda_i - L_T)}}{\Delta \bar{\phi}} - \delta_0 \sum_{i=1}^{k} \frac{1}{(\lambda_i - L_T)(\lambda_i - L_T)} \right) + \frac{1}{\delta \sqrt{r}}
\]

\[
\geq \frac{1}{\delta} - \phi_0 + \mathcal{E}.
\]

Assuming \( \mathcal{E} \geq 0 \) the claim follows immediately because \( \delta_0 = 1 \) and \( \phi_0 = -k / \ell_0 = k / \sqrt{r k} = \sqrt{k / r} \). Thus, we only need to show that \( \mathcal{E} \geq 0 \). From the Cauchy-Schwarz inequality, for \( a_i, b_i \geq 0 \),
To conclude our proof, first note that
\[
\delta_l^{\tau - 1} - \delta_l \phi_r \geq \delta_l^{\tau - 1} - \delta_l \phi_0 = 1 - \sqrt{k/r} > 0 \tag{recall that \(r > k\)}.
\]
Second, \(\lambda_i > L_{\tau + 1}\) because
\[
\lambda_{\min}(A_\tau) > L_\tau + \frac{1}{\phi_r} \geq L_\tau + \frac{1}{\phi_0} = L_\tau + \sqrt{\frac{r}{k}} > L_\tau + 1 = L_{\tau + 1}.
\]
Combining these two observations with eqn. (11) we conclude that \(\mathcal{E} \geq 0\).

Lemma 29 follows from Lemma 31 because the two inequalities must hold simultaneously for at least one index \(j\).

## 8 Dual-set Spectral-Frobenius Sparsification: proof of Lemma 13

In this section we will provide a constructive proof of Lemma 13. Our proof closely follows the proof of Lemma 12, so we will only highlight the differences. We first discuss modifications to Algorithm 1. First of all, the new inputs are \(V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}\) and \(A = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}\). The output is a set of \(n\) non-negative weights \(s_i\), at most \(r\) of which are non-zero. We define the parameters
\[
\delta_l = 1; \quad \delta_u = \frac{\sum_{i=1}^{n} \|a_i\|^2}{1 - \sqrt{\frac{k}{r}}}; \quad L_\tau = \tau - \sqrt{rk}; \quad U_\tau = \tau \delta_u,
\]
for all \(\tau = 0, \ldots, r - 1\). Let \(s_\tau\) denote the vector of weights at the \(\tau\)-th step of Algorithm 1 and initialize \(s_0\) and \(A_0\) as in Algorithm 1 (\(B_0\) will not be necessary). We now define the function \(U_F(\mathbf{a}, \delta_\tau)\), where \(\mathbf{a} \in \mathbb{R}^d\) and \(\delta_\tau \in \mathbb{R}\):
\[
U_F(\mathbf{a}, \delta_\tau) = \delta_u^{-1} a^T a. \tag{12}
\]
Then, at the \(\tau\)-th step, the algorithm will pick an index \(j\) and compute a weight \(t > 0\) such that
\[
U_F(\mathbf{a}_j, \delta_\tau) \leq t^{-1} \leq L(\mathbf{v}_j, \delta_\tau, A_\tau, L_\tau). \tag{13}
\]
The algorithm updates the vector of weights \(s_\tau\) and the matrix
\[
A_\tau = \sum_{i=1}^{n} s_{\tau,i} \mathbf{v}_i \mathbf{v}_i^T.
\]
It is worth noting that the algorithm does not need to update the matrix
\[
B_\tau = \sum_{i=1}^{n} s_{\tau,i} \mathbf{a}_i \mathbf{a}_i^T,
\]
because the function \(U_F\) does not need \(B_\tau\) as input. To prove the correctness of the algorithm we need the following two intermediate lemmas.
Lemma 32. At every step \( \tau = 0, \ldots, r - 1 \) there exists an index \( j \) in \( \{1, \ldots, n\} \) that satisfies eqn. (13).

Proof. The proof is very similar to the proof of Lemma 29 (via Lemma 31) so we only sketch the differences. First, note that the dynamics of \( L \) have not been changed and thus the lower bound for the average of \( L(v_j, \delta_l, A, L) \) still holds. We only need to upper bound the average of \( U_F(a_i, \delta_u) \) as in Lemma 31. Indeed,

\[
\sum_{i=1}^{n} U_F(a_i, \delta_u) = \delta_u^{-1} \sum_{i=1}^{n} a_i^T a_i = \delta_u^{-1} \sum_{i=1}^{n} ||a_i||^2_2 = 1 - \sqrt{\frac{k}{r}},
\]

where the last equality follows from the definition of \( \delta_u \).

Lemma 33. Let \( W \in \mathbb{R}^{\ell \times \ell} \) be a symmetric positive semi-definite matrix, let \( a \in \mathbb{R}^{\ell} \) be a vector, and let \( u \in \mathbb{R} \) satisfy \( u > \text{Tr}(W) \). If \( t > 0 \) satisfies

\[
U_F(a, \delta_u) \leq t^{-1},
\]

then \( \text{Tr}(W + tvv^T) \leq u + \delta_u \).

Proof. Using the conditions of the lemma and the definition of \( U_F \) from eqn. (12),

\[
\text{Tr}(W + tvv^T) - u - \delta_u, = \text{Tr}(W) - u + ta^T a - \delta_u,
\]

\[
\leq \text{Tr}(W) - u < 0,
\]

which concludes the proof of the lemma.

We can now combine Lemmas 27 and 33 to prove that at all steps \( \tau = 0, \ldots, r - 1 \),

\[
\lambda_{\min}(A_\tau) \geq L_\tau \quad \text{and} \quad \text{Tr}(B_\tau) \leq u_\tau.
\]

Note that after all \( r \) steps are completed, \( L_r = r \left( 1 - \sqrt{k/r} \right) \) and \( u_r = r \left( 1 - \sqrt{k/r} \right)^{-1} \sum_{i=1}^{n} ||a_i||^2_2 \). A simple rescaling now concludes the proof. The running time of the (modified) Algorithm 1 is \( O(nr^2 + n\ell) \), where the latter term emerges from the need to compute the function \( U_F(a_j, \delta_u) \) for all \( j = 1, \ldots, n \) once throughout the algorithm.

9 Lower bounds

9.1 Spectral Norm Approximation

Theorem 34. For any \( \alpha > 0 \), any \( k \geq 1 \), and any \( r \geq 1 \), there exists a matrix \( A \in \mathbb{R}^{m \times n} \) for which

\[
\frac{||A - CC^+ A||^2_2}{||A - A_k||^2_2} \geq \frac{n + \alpha^2}{r + \alpha^2}.
\]

Here \( C \) is any matrix that consists of \( r \) columns of \( A \). As \( \alpha \to 0 \), the lower bound is \( n/r \) for the approximation ratio of spectral norm column-based matrix reconstruction.

Proof. We extend the lower bound in [6] to arbitrary \( r > k \). Consider the matrix

\[
A = [e_1 + \alpha e_2, e_1 + \alpha e_3, \ldots, e_1 + \alpha e_{n+1}] \in \mathbb{R}^{(n+1) \times n},
\]

where \( e_i \in \mathbb{R}^{n+1} \) are the standard basis vectors. Then,

\[
A^T A = I_n + \alpha^2 I_n, \quad \sigma^2_1(A) = n + \alpha^2, \quad \text{and} \quad \sigma^2_i(A) = \alpha^2 \quad \text{for} \quad i > 1.
\]
Thus, for all \( k \geq 1 \), \( \| A - A_k \|_2^2 = \alpha^2 \). Intuitively, as \( \alpha \to 0 \), \( A \) is a rank-one matrix. Consider any \( r \) columns of \( A \) and note that, up to row permutations, all sets of \( r \) columns of \( A \) are equivalent. So, without loss of generality, let \( C \) consist of the first \( r \) columns of \( A \). We now compute the optimal reconstruction of \( A \) from \( C \) as follows: let \( a_j \) be the \( j \)-th column of \( A \). In order to reconstruct \( a_j \), we minimize \( \| a_j - Cx \|_2^2 \) over all vectors \( x \in \mathbb{R}^r \). Note that if \( j \leq r \) then the reconstruction error is zero. For \( j > r \), \( a_j = e_1 + \alpha e_{j+1} \),

\[
Cx = e_1 \sum_{i=1}^r x_i + \alpha \sum_{i=1}^r x_i e_{i+1}.
\]

Then,

\[
\| a_j - Cx \|_2^2 = \| e_1 \left( \sum_{i=1}^r x_i - 1 \right) + \alpha \sum_{i=1}^r x_i e_{i+1} - e_{j+1} \|_2^2 = \left( \sum_{i=1}^r x_i - 1 \right)^2 + \alpha^2 \sum_{i=1}^r x_i^2 + 1.
\]

The above quadratic form in \( x \) is minimized when \( x_i = (r + \alpha^2)^{-1} \) for all \( i = 1, \ldots, r \). Let \( \hat{A} = A - CC^+ A \) and let the \( j \)-th column of \( \hat{A} \) be \( \hat{a}_j \). Then, for \( j \leq r \), \( \hat{a}_j \) is an all-zeros vector; for \( j > r \), \( \hat{a}_j = \alpha e_{j+1} - \frac{\alpha}{r+\alpha} \sum_{i=1}^r e_{i+1} \). Thus,

\[
\hat{A}^T \hat{A} = \begin{bmatrix}
0_{r \times r} & 0_{r \times (n-r)} \\
0_{(n-r) \times r} & Z
\end{bmatrix},
\]

where

\[
Z = \frac{\alpha^2}{r+\alpha^2} 1_{n-r} 1_{n-r}^T + \alpha^2 I_{n-r}.
\]

This immediately implies that

\[
\| A - CC^+ A \|_2^2 = \| \hat{A} \|_2^2 = \| \hat{A}^T \hat{A} \|_2 = \| Z \|_2^2 = \frac{(n-r)\alpha^2}{r+\alpha^2} + \alpha^2 = \frac{n+\alpha^2}{r+\alpha^2} \alpha^2.
\]

This concludes our proof, because

\[
\alpha^2 = \| A - A_k \|_2^2.
\]

### 9.2 Frobenius norm approximation

Note that a lower bound for the ratio

\[
\| A - \Pi_{C,k}^\xi (A) \|_\xi^2 / \| A - A_k \|_\xi^2,
\]

does not imply a lower bound for the ratio

\[
\| A - CC^+ A \|_\xi^2 / \| A - A_k \|_\xi^2,
\]

because

\[
\| A - CC^+ A \|_\xi^2 / \| A - A_k \|_\xi^2 \leq \| A - \Pi_{C,k}^\xi (A) \|_\xi^2 / \| A - A_k \|_\xi^2.
\]

Also, note that Proposition 4 in [8] shows a lower bound equal to \( (1 + k/2r) \) for the ratio

\[
\| A - \Pi_{C,k}^F (A) \|_F^2 / \| A - A_k \|_F^2.
\]
For completeness, we extend the bound of [8] to the ratio
\[ \frac{\|A - CC^+A\|^2_F}{\|A - A_k\|^2_F}. \]
In fact, we obtain a lower bound which is asymptotically \( 1 + k/r \). We start with the following lemma.

**Lemma 35.** For any \( \alpha > 0 \) and \( r \geq 1 \), there exists a matrix \( A \in \mathbb{R}^{m \times n} \) for which
\[ \frac{\|A - CC^+A\|^2_F}{\|A - A_1\|^2_F} \geq \frac{n-r}{n-1} \left( 1 + \frac{1}{r + \alpha^2} \right). \]

**Proof.** We employ the same matrix \( A \) as in Theorem 34. So, it follows that
\[ \|A - CC^+A\|^2_F = \text{Tr}(Z) = \alpha^2 (n - r) \left( 1 + \frac{1}{r + \alpha^2} \right), \]
and \( \|A - A_1\|^2_F = (n - 1)\alpha^2 \), which concludes the proof. \( \blacksquare \)

**Theorem 36.** For any \( \alpha > 0, \) any \( k \geq 1 \), and any \( r \geq 1 \), there exists a matrix \( B \in \mathbb{R}^{m \times n} \) for which
\[ \frac{\|B - CC^+B\|^2_F}{\|B - B_k\|^2_F} \geq \frac{n-r}{n-k} \left( 1 + \frac{k}{r + \alpha^2} \right). \]
Here \( C \) is any matrix that consists of \( r \) columns of \( B \). As \( \alpha \to 0 \) and \( n \to \infty \) the lower bound is \( 1 + (k/r) \) for the approximation ratio of Frobenius norm column-based matrix reconstruction.

**Proof.** The matrix \( B \) is constructed as follows. Let \( A \) has dimensions \( \frac{n+1}{k} \times \frac{n}{k} \) and is constructed as in Theorem 34. \( B \) contains \( k \) copies of \( A \) across the main diagonal. We sample \( r \) columns in total, with \( r_i \) columns from each block. Lemma 35 holds for each block, with \( r \) replaced by \( n/k \) and \( r_i \). Let \( r_i \) be the number of columns selected in each block and let \( r = \sum_{i=1}^{k} r_i \). We analyze the Frobenius norm error in each block independently. Let \( Z_i \) be the error matrix in each block, as in the proof of Theorem 34. Then, using Lemma 35, the approximation error is equal to
\[ \|B - CC^+B\|^2_F = \sum_{i=1}^{k} \text{Tr}(Z_i) = \alpha^2 \sum_{i=1}^{k} \left( \frac{n}{k} - r_i \right) \left( 1 + \frac{1}{r_i + \alpha^2} \right). \]
Minimizing this expression subject to the constraint that \( \sum_{i=1}^{k} r_i = r \) gives \( r_i = r/k \). The result follows after some algebra using \( \|B - B_k\|^2_F = (n - k)\alpha^2 \). \( \blacksquare \)

### 10 Open Problems

Several interesting questions remain unanswered; we highlight two. First, is it possible to improve the running time of the deterministic algorithms of Lemmas 12 and 13? Recently, Zouzias [25] made progress in improving the running time of the spectral sparsification result of [1]; can we get a similar improvement for the 2-set algorithms presented here? Second, in the parlance of Theorem 5, is there a deterministic algorithm that selects \( O(k/\epsilon) \) columns from \( A \) and guarantees relative-error accuracy for the error \( \|A - \Pi^k_{C_{\epsilon}}(A)\|^2_F \)? In a very recent development, [18] partially answers this question by extending the volume sampling approach of [6] to deterministically select \( k(1 + o(1)) \) columns and obtain a relative error bound for the term \( \|A - CC^+A\|^2_F \). Notice that it is not obvious if [18] implies a similar deterministic bound for the error \( \|A - \Pi^k_{C_{\epsilon}}(A)\|^2_F \).
Acknowledgments

We would like to thank A. Deshpande, D. Feldman, K. Varadarajan, and J. Tropp for useful discussions. We also thank D. Feldman and K. Varadarajan for pointing out the connections between the subspace approximation line of research [7, 13, 14, 24] and our work. This work has been supported by NSF CCF-1016501 and NSF DMS-1008983.

References

[1] J.D. Batson, D.A. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. In Proc. 41st STOC, pages 255–262, 2009.

[2] H. Avron and C. Boutsidis Faster Subset Selection for Matrices and Applications. arXiv report: arXiv:1201.0127v1, http://arxiv.org/abs/1201.0127, 2012.

[3] C. Boutsidis, M.W. Mahoney, and P. Drineas An improved approximation algorithm for the column subset selection problem. In Proc. 20th SODA, 2009.

[4] T.F. Chan and P.C. Hansen. Some applications of the rank revealing QR factorization. SIAM Journal on Scientific and Statistical Computing, 13:727–741, 1992.

[5] K.L. Clarkson and D.P. Woodruff. Numerical linear algebra in the streaming model. In Proc. 41st STOC, pages 205–214, 2009.

[6] A. Deshpande and L. Rademacher. Efficient volume sampling for row/column subset selection. In Proc. 42th STOC, pages 329–338, 2010.

[7] A. Deshpande and K. R. Varadarajan. Sampling-based dimension reduction for subspace approximation. In Proc. 39th STOC, pages 641–650, 2007.

[8] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. In Proc. of RANDOM - APPROX, 2006.

[9] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. Theory of Computing, 2(12):225–247, 2006.

[10] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In Proc. 10th SODA, pages 291–299, 1999.

[11] P. Drineas, I. Kerenidis, and P. Raghavan. Competitive recommendation systems. In Proc. 34th STOC, pages 82–90, 2002.

[12] P. Drineas, M.W. Mahoney, and S. Muthukrishnan. Polynomial time algorithm for column-row based relative-error low-rank matrix approximation. Technical Report 2006-04, DIMACS, March 2006.

[13] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In Proc. 43rd STOC, 2011.

[14] D. Feldman, M. Monemizadeh, C. Sohler, and D. Woodruf. Coresets and sketches for high dimensional subspace approximation problems. In Proc. 21st SODA, pages 630–649, 2010.

[15] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In Proc. 39th FOCS, pages 370–378, 1998.
[16] G. Golub. Numerical methods for solving linear least squares problems. *Numerische Mathematik*, 7:206–216, 1965.

[17] M. Gu and S.C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17:848–869, 1996.

[18] V. Guruswami and A. K. Sinop. Optimal column-based low-rank matrix reconstruction. In *Proc. 23rd SODA*, 2012.

[19] N. Halko, P.G. Martinsson, and J.A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 2011.

[20] E. Liberty, F. Woolfe, P.G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007.

[21] M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106:697–702, 2009.

[22] V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100–1124, 2009.

[23] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *Proc. 47th FOCS*, pages 143–152, 2006.

[24] N. D. Shyamalkumar and K. R. Varadarajan. Efficient subspace approximation algorithms. In *Proc. 18th SODA*, pages 532–540, 2007.

[25] A. Zouzias. arXiv report: arXiv:1103.2793v1, http://arxiv.org/abs/1103.2793, 2011.