LazySVD: Even Faster SVD Decomposition
Yet Without Agonizing Pain

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Institute for Advanced Study & Princeton University

Yuanzhi Li
yuanzhil@cs.princeton.edu
Princeton University

first circulated on May 20, 2016

Abstract

We study $k$-SVD that is to obtain the first $k$ singular vectors of a matrix $A$. Recently, a few breakthroughs have been discovered on $k$-SVD: Musco and Musco [19] proved the first gap-free convergence result using the block Krylov method, Shamir [21] discovered the first variance-reduction stochastic method, and Bhojanapalli et al. [7] provided the fastest $O(\text{nnz}(A) + \text{poly}(1/\varepsilon))$-time algorithm using alternating minimization.

In this paper, we put forward a new and simple LazySVD framework to improve the above breakthroughs. This framework leads to a faster gap-free method outperforming [19], and the first accelerated and stochastic method outperforming [21]. In the $O(\text{nnz}(A) + \text{poly}(1/\varepsilon))$ running-time regime, LazySVD outperforms [7] in certain parameter regimes without even using alternating minimization.

1 Introduction

The singular value decomposition (SVD) of a rank-$r$ matrix $A \in \mathbb{R}^{d \times n}$ corresponds to decomposing $A = V \Sigma U^\top$ where $V \in \mathbb{R}^{d \times r}$, $U \in \mathbb{R}^{n \times r}$ are two column orthonormal matrices, and $\Sigma = \text{diag}\{\sigma_1, \ldots, \sigma_r\} \in \mathbb{R}^{r \times r}$ is a non-negative diagonal matrix with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$. The columns of $V$ (resp. $U$) are called the left (resp. right) singular vectors of $A$ and the diagonal entries of $\Sigma$ are called the singular values of $A$. SVD is one of the most fundamental tools used in machine learning, computer vision, statistics, and operations research, and is essentially equivalent to principal component analysis (PCA) up to column averaging.

A rank $k$ partial SVD, or $k$-SVD for short, is to find the top $k$ left singular vectors of $A$, or equivalently, the first $k$ columns of $V$. Denoting by $V_k \in \mathbb{R}^{d \times k}$ the first $k$ columns of $V$, and $U_k$ the first $k$ columns of $U$, one can define $A_k^* \overset{\text{def}}{=} V_k V_k^T A = V_k \Sigma_k U_k^T$ where $\Sigma_k = \text{diag}\{\sigma_1, \ldots, \sigma_k\}$. Under this notation, $A_k^*$ is the the best rank-$k$ approximation of matrix $A$ in terms of minimizing $\|A - A_k\|$ among all rank-$k$ matrices $A_k$. Here, the norm can be any Schatten-$q$ norm for $q \in [1, \infty]$, including spectral norm ($q = \infty$) and Frobenius norm ($q = 2$), therefore making $k$-SVD a very powerful tool for information retrieval, data de-noising, or even data compression.

Traditional algorithms to compute SVD essentially run in time $O(nd \min\{d, n\})$, which is usually very expensive for big-data scenarios. As for $k$-SVD, defining $\text{gap} \overset{\text{def}}{=} (\sigma_k - \sigma_{k+1})/\sigma_k$ to be the relative $k$-th eigengap of matrix $A$, the famous subspace power method or block Krylov method [14] solves $k$-SVD in time $O(\text{gap}^{-1} \cdot k \cdot \text{nnz}(A) \cdot \log(1/\varepsilon))$ or $O(\text{gap}^{-0.5} \cdot k \cdot \text{nnz}(A) \cdot \log(1/\varepsilon))$ respectively if ignoring lower-order terms. Here, $\text{nnz}(A)$ is the number of non-zero elements in matrix $A$, and the more precise running times are stated in Table 1.
Recently, there are breakthroughs to compute k-SVD faster, from three distinct perspectives.

The first breakthrough is the work of Musco and Musco [19] for proving a running time for k-SVD that does not depend on singular value gaps (or any other properties) of $A$. As highlighted in [19], providing gap-free results was an open question for decades and is a more reliable goal for practical purposes. Specifically, they proved that the block Krylov method converges in time $O\left(\frac{\nnz(A)}{\varepsilon^2} + \frac{k^2d}{\varepsilon} + \frac{k^3}{\varepsilon^{3/2}}\right)$, where $\varepsilon$ is the multiplicative approximation error \(^1\).

The second breakthrough is the work of Shamir [21] for providing a fast stochastic k-SVD algorithm. In a stochastic setting, one assumes\(^2\)

$$A = \frac{1}{n} \sum_{i=1}^{n} a_i a_i^\top$$

instead of repeatedly multiplying matrix $AA^\top$ to a vector in the (subspace) power method, Shamir proposed to use a random rank-1 copy $a_i a_i^\top$ to approximate such multiplications. When equipped with very ad-hoc variance-reduction techniques, Shamir showed that the algorithm has a better (local) performance than power method (see Table 1). Unfortunately, Shamir’s result is (1) not gap-free; (2) not accelerated (i.e., does not match the $\varepsilon^{-0.5}$ dependence comparing to block Krylov); and (3) requires a very accurate warm-start that in principle can take a very long time to compute.

The third breakthrough is in obtaining running times of the form $O(\nnz(A) + \poly(k, 1/\varepsilon) \cdot (n + d))$ [7, 8], see Table 2. We call them NNZ results. To obtain NNZ results, one needs sub-sampling

\(^1\)In this paper, we use $\tilde{O}$ notations to hide possible logarithmic factors on $1/\varepsilon, n, d, k$ and potentially also on $\sigma_k/\sigma_{k+1}$.

\(^2\)This normalization follows the tradition of stochastic $k$-SVD or 1-SVD literatures [12, 20, 21] in order to state results more cleanly.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Running time $(\times$ for being outperformed)</th>
<th>GF?</th>
<th>Stoc?</th>
<th>Acc?</th>
</tr>
</thead>
<tbody>
<tr>
<td>subspace PM [19]</td>
<td>$O\left(\frac{\nnz(A)}{\varepsilon} + \frac{k^2d}{\varepsilon}\right)$</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>$O\left(\frac{\nnz(A)}{\gap} + \frac{k^2d}{\gap}\right)$</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>block Krylov [19]</td>
<td>$\tilde{O}\left(\frac{\nnz(A)}{\varepsilon^{1/2}} + \frac{k^2d}{\varepsilon} + \frac{k^3}{\varepsilon^{3/2}}\right)$</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>$\tilde{O}\left(\frac{\nnz(A)}{\gap^{1/2}} + \frac{k^2d}{\gap^{1/2}} + \frac{k^3}{\gap^{3/2}}\right)$</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>LazySVD Corollary 4.3 and 4.4</td>
<td>$\tilde{O}\left(\frac{\nnz(A)}{\varepsilon^{1/2}} + \frac{k^2d}{\varepsilon} + \frac{k^3}{\varepsilon^{3/2}}\right)$</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>$\tilde{O}\left(\frac{\nnz(A)}{\gap^{1/2}} + \frac{k^2d}{\gap^{1/2}} + \frac{k^3}{\gap^{3/2}}\right)$</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Shamir [21]</td>
<td>$\tilde{O}(knd + \frac{k^2d}{\sigma_k^{1/2}})$ (local convergence only)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>LazySVD Corollary 4.3 and 4.4</td>
<td>$\tilde{O}\left(\frac{\nnz(A)}{\varepsilon^{1/2}} + \frac{k^2d}{\varepsilon} + \frac{k^3}{\varepsilon^{3/2}}\right)$</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>$\tilde{O}\left(\frac{\nnz(A)}{\gap^{1/2}} + \frac{k^2d}{\gap^{1/2}} + \frac{k^3}{\gap^{3/2}}\right)$</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

All GF results above provide $(1 + \varepsilon)\||\Delta||_2$ spectral and $(1 + \varepsilon)\||\Delta||_F$ Frobenius guarantees.

Table 1: Performance comparison among direct methods. Define $\gap = (\sigma_k - \sigma_{k+1})/\sigma_k \in [0, 1]$. GF = Gap Free; Stoc = Stochastic; Acc = Accelerated. Stochastic results in this table are assuming $\|a_i\|_2 \leq 1$ following (1.1).
on the matrix and this incurs a poor dependence on \( \varepsilon \). For this reason, the polynomial dependence on \( 1/\varepsilon \) is usually considered as the most important factor. In 2015, Bhojanapalli et al. [7] obtained a \( 1/\varepsilon^2 \)-rate NNZ result using alternating minimization. Since \( 1/\varepsilon^2 \) also shows up in the sampling complexity, we believe the quadratic dependence on \( \varepsilon \) is tight among NNZ types of algorithms.

All the cited results rely on ad-hoc non-convex optimization techniques together with matrix algebra, which make the final proofs complicated. Furthermore, Shamir’s result [21] only works if a \( 1/poly(d) \)-accurate warm start is given, and the time needed to find a warm start is unclear.

In this paper, we develop a new algorithmic framework to solve \( k \)-SVD. It not only improves the aforementioned breakthroughs, but also relies only on simple convex analysis.

### 1.1 Our Results and the Settlement of an Open Question

We propose to use an extremely simple framework that we call LazySVD to solve \( k \)-SVD:

\[
\text{LazySVD: perform 1-SVD repeatedly, \( k \) times in total.}
\]

More specifically, in this framework we first compute the leading singular vector \( v \) of \( A \), and then left-project \( (I - vv^\top)A \) and repeat this procedure \( k \) times. Quite surprisingly,

This seemingly “most-intuitive” approach was widely considered as “not a good idea.”

In textbooks and research papers, one typically states that LazySVD has a running time that inversely depends on all the intermediate singular value gaps \( \sigma_1 - \sigma_2, \ldots, \sigma_k - \sigma_{k+1} \) [18, 21]. This dependence makes the algorithm useless if some singular values are close, and is even thought to be necessary [18]. For this reason, textbooks describe only block methods (such as block power method, block Krylov, alternating minimization) which find the top \( k \) singular vectors together. Musco and Musco [19] stated as an open question to design “single-vector” methods without running time dependence on all the intermediate singular value gaps.

In this paper, we fully answer this open question with novel analyses on this LazySVD framework. In particular, the resulting running time either

- depends on \( \text{gap}^{-0.5} \) where \( \text{gap} \) is the relative singular value gap only between \( \sigma_k \) and \( \sigma_{k+1} \), or
- depends on \( \varepsilon^{-0.5} \) where \( \varepsilon \) is the approximation ratio (so is gap-free).

Such dependency matches the best known dependency for block methods.

More surprisingly, by making different choices of the 1-SVD subroutine in this LazySVD framework, we obtain multiple algorithms for different needs (see Table 1 and 2):

- If accelerated gradient descent or Lanczos algorithm is used for 1-SVD, we obtain a faster \( k \)-SVD algorithm than block Krylov [19].
- If a variance-reduction stochastic method is used for 1-SVD, we obtain the first accelerated stochastic algorithm for \( k \)-SVD, and this outperforms Shamir [21].
- If one sub-samples \( A \) before applying LazySVD, the running time becomes \( \tilde{O}(\text{nnz}(A) + \varepsilon^{-2}poly(k) \cdot d) \). This improves upon [7] in certain (but sufficiently interesting) parameter regimes, but completely avoids the use of alternating minimization.

Finally, besides the running time advantages above, our analysis is completely based on convex optimization because 1-SVD is solvable using convex techniques. LazySVD also works when \( k \) is not known to the algorithm, as opposed to block methods which need to know \( k \) in advance.
Other Related Work. Some authors focus on the streaming or online model of 1-SVD [4, 15, 17] or $k$-SVD [3]. These algorithms are slower than offline methods. Unlike $k$-SVD, accelerated stochastic methods were previously known for 1-SVD [12, 13]. After this paper is published, LazySVD has been generalized to also solve canonical component analysis and generalized PCA by the same authors [1]. If one is only interested in projecting a vector to the top $k$-eigenspace without computing the top $k$ eigenvectors like we do in this paper, this can also be done in an accelerated manner [2].

2 Preliminaries

Given a matrix $A$ we denote by $\|A\|_2$ and $\|A\|_F$ respectively the spectral and Frobenius norms of $A$. For $q \geq 1$, we denote by $\|A\|_{S_q}$ the Schatten $q$-norm of $A$. We write $A \succeq B$ if $A, B$ are symmetric and $A - B$ is positive semi-definite (PSD). We denote by $\lambda_k(M)$ the $k$-th largest eigenvalue of a symmetric matrix $M$, and $\sigma_k(A)$ the $k$-th largest singular value of a rectangular matrix $A$.

Since $\lambda_k(AA^\top) = \lambda_k(A^\top A) = (\sigma_k(A))^2$,

solving $k$-SVD for $A$ is the same as solving $k$-PCA for $M = AA^\top$.

We denote by $\sigma_1 \geq \ldots \sigma_d \geq 0$ the singular values of $A \in \mathbb{R}^{d \times n}$, by $\lambda_1 \geq \ldots \lambda_d \geq 0$ the eigenvalues of $M = AA^\top \in \mathbb{R}^{d \times d}$. (Although $A$ may have fewer than $d$ singular values for instance when $n < d$, if this happens, we append zeros.) We denote by $A_k^*$ the best rank-$k$ approximation of $A$.

We use $\perp$ to denote the orthogonal complement of a matrix. More specifically, given a column orthonormal matrix $U \in \mathbb{R}^{d \times k}$, we define $U^{\perp} \overset{\text{def}}{=} \{x \in \mathbb{R}^d \mid U^\top x = 0\}$. For notational simplicity, we sometimes also denote $U^{\perp}$ as a $d \times (d - k)$ matrix consisting of some basis of $U^{\perp}$.

**Theorem 2.1** (approximate matrix inverse). Given $d \times d$ matrix $M \succeq 0$ and constants $\lambda, \delta > 0$ satisfying $\lambda I - M \succeq \delta I$, one can minimize the quadratic $f(x) \overset{\text{def}}{=} x^\top (\lambda I - M)x$ in order to invert $(\lambda I - M)^{-1}b$. Suppose the desired accuracy is $\|x - (\lambda I - M)^{-1}b\| \leq \varepsilon$. Then,

- Accelerated gradient descent (AGD) produces such an output $x$ in $O(\frac{\lambda^{1/2}}{\delta^{1/4}} \log \frac{\lambda}{\varepsilon^2})$ iterations, each requiring $O(d)$ time plus the time needed to multiply $M$ with a vector.
- If $M$ is given in the form $M = \frac{1}{d} \sum_{i=1}^n a_i a_i^\top$ and $\|a_i\|_2 \leq 1$, then accelerated SVRG (see for instance [5]) produces such an output $x$ in time $O(\max\{nd, \frac{n^{3/4}d \lambda^{1/4}}{\delta^{1/2}}\} \log \frac{\lambda}{\varepsilon^2})$. 

<table>
<thead>
<tr>
<th>Paper</th>
<th>Running time</th>
<th>Frobenius norm</th>
<th>Spectral norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>[8]</td>
<td>$O(\text{nnz}(A)) + O\left(\frac{k^2}{\varepsilon^2}(n + d) + \frac{k}{\varepsilon}\right)$</td>
<td>$(1 + \varepsilon)|\Delta|_F$</td>
<td>$(1 + \varepsilon)|\Delta|_F$</td>
</tr>
<tr>
<td>[7]</td>
<td>$O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^2(\sigma_1/\sigma_k + 1)^2}{\varepsilon^2}(n + d)\right)$</td>
<td>$(1 + \varepsilon)|\Delta|_F$</td>
<td>$|\Delta|_2 + \varepsilon|\Delta|_F$</td>
</tr>
<tr>
<td>LazySVD</td>
<td><strong>Theorem 5.1</strong></td>
<td>$O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^2}{\varepsilon}(\sigma_1/\sigma_k + 1)^2 \cdot d\right)$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^2}{\varepsilon}(\sigma_1/\sigma_k + 1)^2 \cdot d\right)$</td>
<td>N/A</td>
<td>$|\Delta|_2 + \varepsilon|\Delta|_F$</td>
</tr>
<tr>
<td></td>
<td>$O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^2}{\varepsilon}(\sigma_1/\sigma_k + 1)^2 \cdot d\right)$</td>
<td>N/A</td>
<td>$|\Delta|_2 + \varepsilon|\Delta|_F$</td>
</tr>
</tbody>
</table>

Table 2: Performance comparison among $O(\text{nnz}(A) + \text{poly}(1/\varepsilon))$ type of algorithms. Remark: we have not tried hard to improve the dependency with respect to $k$ or $(\sigma_1/\sigma_{k+1})$. See Remark 5.2.
Algorithm 1 \text{AppxPCA}(\mathcal{A}, M, \delta_x, \varepsilon, p) \quad \diamond \text{(only for proving our theoretical results; for practitioners, feel free to use your favorite 1-PCA algorithm such as Lanczos to replace \text{AppxPCA})}

\textbf{Input: } \mathcal{A}, \text{ an approximate matrix inversion method}; \ M \in \mathbb{R}^{d \times d}, \text{ a symmetric matrix satisfying } 0 \preceq M \preceq I; \ \delta_x \in (0, 0.5], \text{ a multiplicative error}; \ \varepsilon \in (0, 1), \text{ a numerical accuracy parameter; } \\
\text{ and } p \in (0, 1), \text{ a confidence parameter.} \quad \diamond \text{ running time only logarithmically depends on } 1/\varepsilon \text{ and } 1/p.

1: \quad m_1 \leftarrow \left\lfloor 4 \log \left( \frac{286d}{p\varepsilon} \right) \right\rfloor, \ m_2 \leftarrow \left\lfloor \log \left( \frac{36d}{p\varepsilon} \right) \right\rfloor;
\quad \diamond \ m_1 = T^{\text{PM}}(8, 1/32, p) \text{ and } m_2 = T^{\text{PM}}(2, \varepsilon/4, p) \text{ using definition in Lemma A.1}
2: \quad \tilde{\varepsilon}_1 \leftarrow \frac{1}{64m_1} \left( \frac{\delta_x}{\varepsilon} \right)^{m_1} \text{ and } \tilde{\varepsilon}_2 \leftarrow \frac{\varepsilon}{8m_2} \left( \frac{\delta_x}{\varepsilon} \right)^{m_2}
3: \quad \tilde{w}_0 \leftarrow \text{a random unit vector}; \ s \leftarrow 0; \ \lambda^{(0)} \leftarrow 1 + \delta_x;
4: \quad \text{repeat}
5: \quad s \leftarrow s + 1;
6: \quad \text{for } t = 1 \text{ to } m_1 \text{ do}
7: \quad \text{Apply } \mathcal{A} \text{ to find } \tilde{w}_t \text{ satisfying } \| \tilde{w}_t - (\lambda^{(s-1)} I - M)^{-1} \tilde{w}_{t-1} \| \leq \tilde{\varepsilon}_1;
8: \quad \text{end for}
9: \quad w \leftarrow \tilde{w}_{m_1} / \| \tilde{w}_{m_1} \|;
10: \quad \text{Apply } \mathcal{A} \text{ to find } v \text{ satisfying } \| v - (\lambda^{(s-1)} I - M)^{-1} w \| \leq \tilde{\varepsilon}_1;
11: \quad \Delta^{(s)} \leftarrow \frac{1}{2} \cdot \frac{1}{w^\top v - \tilde{\varepsilon}_1} \text{ and } \lambda^{(s)} \leftarrow \lambda^{(s-1)} - \frac{\Delta^{(s)}}{2};
12: \quad \text{until } \Delta^{(s)} \leq \frac{\delta_x \lambda^{(s)}}{3}
13: \quad f \leftarrow s;
14: \quad \text{for } t = 1 \text{ to } m_2 \text{ do}
15: \quad \text{Apply } \mathcal{A} \text{ to find } \tilde{w}_t \text{ satisfying } \| \tilde{w}_t - (\lambda^{(f)} I - M)^{-1} \tilde{w}_{t-1} \| \leq \tilde{\varepsilon}_2;
16: \quad \text{end for}
17: \quad \text{return } w \overset{\text{def}}{=} \tilde{w}_{m_2} / \| \tilde{w}_{m_2} \|.

3 \ A \text{ Specific 1-SVD Algorithm: Shift-and-Inverse Revisited}

In this section, we study a specific 1-PCA algorithm \text{AppxPCA} (recall 1-PCA equals 1-SVD). It is a (multiplicative-)approximate algorithm for computing the leading eigenvector of a symmetric matrix.

We emphasize that, in principle, most known 1-PCA algorithms (e.g., power method, Lanczos method) are suitable for our LazySVD framework. We choose \text{AppxPCA} solely because it provides the maximum flexibility in obtaining all stochastic / NNZ running time results at once.

Our \text{AppxPCA} uses the shift-and-inverse routine [12, 13], and our pseudo-code in Algorithm 1 is a modification of Algorithm 5 that appeared in [12]. Since we need a more refined running time statement with a multiplicative error guarantee, and since the stated proof in [12] is anyways only a sketched one, we choose to carefully reprove a similar result of [12] (details in Appendix A) and state the following theorem:

\textbf{Theorem 3.1 (AppxPCA). Let } M \in \mathbb{R}^{d \times d} \text{ be a symmetric matrix with eigenvalues } 1 \geq \lambda_1 \geq \cdots \geq \lambda_d \geq 0 \text{ and corresponding eigenvectors } u_1, \ldots, u_d \text{. With probability at least } 1 - p, \text{ \text{AppxPCA} produces an output } w \text{ satisfying}

\[ \sum_{i \in [d], \lambda_i \leq (1 - \delta_x) \lambda_1} (w^\top u_i)^2 \leq \varepsilon \text{ and } w^\top M w \geq (1 - \delta_x)(1 - \varepsilon) \lambda_1 \].

Furthermore, the total number of oracle calls to } \mathcal{A} \text{ is } O(\log(1/\delta_x)m_1 + m_2), \text{ and each time we call } \mathcal{A} \text{ we have } \frac{\lambda^{(s)}}{\lambda_{\min}(\lambda^{(s)} I - M)} \leq \frac{12}{\delta_x} \text{ and } \frac{1}{\lambda_{\min}(\lambda^{(s)} I - M)} \leq \frac{12}{\delta_x \lambda_1}.\]
Algorithm 2 LazySVD(A, M, k, δ, ε_{pca}, p)

Input: A, an approximate matrix inversion method; M ∈\mathbb{R}^{d \times d}, a matrix satisfying 0 ≤ M ≤ I; k ∈ [d], the desired rank; δ ∈ (0, 1), a multiplicative error; ε_{pca} ∈ (0, 1), a numerical accuracy parameter; and p ∈ (0, 1), a confidence parameter.

1: M_0 ← M and V_0 ← []
2: for s = 1 to k do
3: \( v'_s \) ← AppxPCA(A, M_{s-1}, δ/2, ε_{pca}, p/k);
4: \( v_s \) ← \((I - V_{s-1}V_{s-1}^\top)v'_s)/(\|I - V_{s-1}V_{s-1}^\top\|v'_s); \)
5: \( V_s \) ← [V_{s-1}, v_s];
6: \( M_s \) ← (I - v_sv_s^\top)M_{s-1}(I - v_sv_s^\top)

end for
8: return V_k.

Since AppxPCA reduces 1-PCA to oracle calls of a matrix inversion subroutine A, the stated conditions \( \frac{\lambda_i}{\lambda(\langle\rangle - M)} ≤ \frac{12}{d} \) and \( \frac{1}{\lambda_{\min}(A^\top M)} ≤ \frac{12}{d} \) in Theorem 3.1, together with complexity results for matrix inversions (see Theorem 2.1), imply the following running times for AppxPCA.

Corollary 3.2.

- If A is AGD, the running time of AppxPCA is \( \tilde{O}\left(\frac{1}{d^{1/2}}\right) \) multiplied with O(d) plus the time needed to multiply M with a vector.
- If \( M = \frac{1}{n} \sum_{i=1}^{n} a_i a_i^\top \) where each \( \|a_i\|_2 ≤ 1 \), and A is accelerated SVRG, then the total running time of AppxPCA is \( \tilde{O}\left(\max\{nd, \frac{n^{3/4}}{\sqrt{d}}, \frac{1}{\sqrt{d}}\}\right) \).

4 Main Algorithm and Theorems

Our algorithm LazySVD is stated in Algorithm 2. It starts with \( M_0 = M \) and repeatedly applies k times AppxPCA. In the s-th iteration, it computes an approximate leading eigenvector of matrix \( M_{s-1} \) using AppxPCA with a multiplicative error δ/2, projects \( M_{s-1} \) to the orthogonal space of this vector, and then calls it matrix \( M_s \).

In this stated form, LazySVD finds approximately the top k eigenvectors of a symmetric matrix \( M ∈ \mathbb{R}^{d \times d} \). If M is given as \( M = AA^\top \), then LazySVD automatically finds the k-SVD of A.

4.1 Our Core Theorems

We state our approximation and running time core theorems of LazySVD below, and then provide corollaries to translate them into gap-dependent and gap-free theorems on k-SVD.

Theorem 4.1 (approximation). Let \( M ∈ \mathbb{R}^{d \times d} \) be a symmetric matrix with eigenvalues \( 1 ≥ \lambda_1 ≥ \cdots ≥ \lambda_d \geq 0 \) and corresponding eigenvectors \( u_1, \ldots, u_d \). Let \( k ∈ [d] \), let δ, p ∈ (0, 1), and let \( ε_{pca} ≤ poly(ε, δ, 1/\sqrt{d}, \lambda, 1/\lambda) \). Then, LazySVD outputs a (column) orthonormal matrix \( V_k = (v_1, \ldots, v_k) ∈ \mathbb{R}^{d \times k} \) which, with probability at least \( 1 - p \), satisfies all of the following properties. (Denote by \( M_k = (I - V_kV_k^\top)M(I - V_kV_k^\top) \).)

The detailed specifications of \( ε_{pca} \) can be found in the appendix where we restate the theorem more formally. To provide the simplest proof, we have not tightened the polynomial factors in the theoretical upper bound of \( ε_{pca} \) because the running time depends only logarithmic on \( 1/ε_{pca} \).
(a) Core lemma: \[ \|V_k^T U\|_2 \leq \epsilon, \] where \( U = (u_j, \ldots, u_d) \) is the (column) orthonormal matrix and \( j \) is the smallest index satisfying \( \lambda_j \leq (1 - \delta_x)\|M_{k-1}\|_2 \).

(b) Spectral norm guarantee: \( \lambda_{k+1} \leq \|M_k\|_2 \leq \frac{\lambda_{k+1}}{1 - \delta_x} \).

(c) Rayleigh quotient guarantee: \( (1 - \delta_x)\lambda_k \leq v_k^T M v_k \leq \frac{1}{1 - \delta_x} \lambda_k \).

(d) Schatten-q norm guarantee: for every \( q \geq 1 \), we have \( \|M_k\|_{S_q} \leq \frac{(1 + \delta_x)^2}{(1 - \delta_x)^2} \left( \sum_{i=k+1}^d \lambda_i^q \right)^{1/q} \).

We defer the proof of Theorem 4.1 to the appendix, but highlight the main ideas and techniques behind the proof in Section 4.3. Below we state the running time of \texttt{LazySVD}.

**Theorem 4.2** (running time). \texttt{LazySVD} can be implemented to run in time

- \( \tilde{O}(\frac{\text{nnz}(M) + k^2 d}{\delta_x^2}) \) if \( A \) is AGD and \( M \in \mathbb{R}^{d \times d} \) is given explicitly;
- \( \tilde{O}(\frac{\text{nnz}(A) + k^2 d}{\delta_x^2}) \) if \( A \) is AGD and \( M \) is given as \( M = A A^\top \) where \( A \in \mathbb{R}^{d \times n} \); or
- \( \tilde{O}(k^d + \frac{k d^3/4 d}{\lambda_k^3/\delta_x^3}) \) if \( A \) is accelerated SVRG and \( M = \frac{1}{n} \sum_{i=1}^n a_i a_i^\top \) where each \( \|a_i\|_2 \leq 1 \).

Above, the \( \tilde{O} \) notation hides logarithmic factors with respect to \( k, d, 1/\delta_x, 1/p, 1/\lambda_1, \lambda_1/\lambda_k \).

**Proof of Theorem 4.2** We call \( k \) times \texttt{AppxPCA} and each time we can feed \( M_{s-1} = (I - V_{s-1} V_{s-1}^\top) M (I - V_{s-1} V_{s-1}^\top) \) implicitly into \texttt{AppxPCA} thus the time needed to multiply \( M_{s-1} \) with a \( d \)-dimensional vector is \( O(d k + \text{nnz}(M)) \) or \( O(d k + \text{nnz}(A)) \). Here, the \( O(d k) \) overhead is due to the projection of a vector into \( V_{s-1}^\top \). This proves the first two running times using Corollary 3.2.

To obtain the third running time, when we compute \( M_s \) from \( M_{s-1} \), we explicitly project \( a_i' \leftarrow (I - v_s v_s^\top) a_i \) for each vector \( a_i \), and feed the new \( a_1', \ldots, a_n' \) into \texttt{AppxPCA}. Now the running time follows from the second part of Corollary 3.2 together with the fact that \( \|M_{s-1}\|_2 \geq \|M_{k-1}\|_2 \geq \lambda_k \). \( \square \)

### 4.2 Our Main Results for \( k \)-SVD

Our main theorems imply the following corollaries (proved in Appendix C.1 for completeness).

**Corollary 4.3** (Gap-dependent \( k \)-SVD). Let \( A \in \mathbb{R}^{d \times n} \) be a matrix with singular values \( 1 \geq \sigma_1 \geq \cdots \sigma_d \geq 0 \) and the corresponding left singular vectors \( u_1, \ldots, u_d \in \mathbb{R}^d \). Let \( \text{gap} = \frac{\sigma_k - \sigma_{k+1}}{\sigma_k} \) be the relative gap. For fixed \( \epsilon, p > 0 \), consider the output

\[ V_k \leftarrow \texttt{LazySVD}(A, AA^\top, k, \text{gap}, O(\frac{\epsilon^4 \text{gap}^2}{k^4 (\sigma_1/\sigma_k)^2}), p) \text{ .} \]

Then, defining \( W = (u_{k+1}, \ldots, u_d) \), we have with probability at least \( 1 - p \):

\[ V_k \text{ is a rank-} k \ (\text{column}) \text{ orthonormal matrix with } \|V_k^T W\|_2 \leq \epsilon \text{ .} \]

Our running time is \( \tilde{O}(\frac{\text{nnz}(A) + k^2 d}{\sqrt{\text{gap}}} \sqrt{\text{gap}}) \), or time \( \tilde{O}(k^d + \frac{k d^3/4 d}{\lambda_k^3/\delta_x^3}) \) in the stochastic setting \( (1.1) \).

Above, both running times depend only poly-logarithmically on \( 1/\epsilon \).
Corollary 4.4 (Gap-free k-SVD). Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $1 \geq \sigma_1 \geq \cdots \sigma_d \geq 0$. For fixed $\varepsilon, p > 0$, consider the output

$$(v_1, \ldots, v_k) = V_k \leftarrow \text{LazySVD}(A, A^T, k, \varepsilon, \mathcal{O}\left(\frac{\varepsilon^6}{k^4d^4(\sigma_1/\sigma_{k+1})^2}\right), p).$$

Then, defining $A_k = V_k V_k^T A$ which is a rank $k$ matrix, we have with probability at least $1 - p$:

1. **Spectral norm guarantee**: $\|A - A_k\|_2 \leq (1 + \varepsilon)\|A - A^*_k\|_2$;
2. **Frobenius norm guarantee**: $\|A - A_k\|_F \leq (1 + \varepsilon)\|A - A^*_k\|_F$; and
3. **Rayleigh quotient guarantee**: $\forall i \in [k], |v_i^T A A^T v_i - \sigma_i^2| \leq \varepsilon \sigma_i^2$.

Running time is $\tilde{O}\left(\frac{knz(A) + k^2 d}{\varepsilon^2}\right)$, or time $\tilde{O}(kn + \frac{k^{3/4} d}{\sigma_k^2 \varepsilon^2})$ in the stochastic setting.\cite{11}

Remark 4.5. The spectral and Frobenius guarantees we adopted are standard. It was observed that the spectral guarantee is more desirable than the Frobenius one in practice.\cite{19}. In fact, our algorithm implies for all $q \geq 1, \|A - A_k\|_{S_q} \leq (1 + \varepsilon)\|A - A^*_k\|_{S_q}$ where $\|\cdot\|_{S_q}$ is the Schatten-$q$ norm. Rayleigh-quotient types of guarantee were introduced by Musco and Musco\cite{19} for a more refined comparison. They showed that the block Krylov method satisfies $|v_i^T A A^T v_i - \sigma_i^2| \leq \varepsilon \sigma_{k+1}^2$, which is slightly stronger than ours. However, these two guarantees are not much different in practice as we evidenced in our experiments.

4.3 High-Level Ideas Behind Our Theorems

For the sake of demonstrating the idea, we focus on the case when there is a (known) relative gap $\text{gap} \overset{\text{def}}{=} (\sigma_k - \sigma_{k+1})/\sigma_k \in [0, 1]$ between the $k$-th and the $(k + 1)$-th singular values of $A$. Note that LazySVD can be equivalently viewed as follows. At iteration $s$, LazySVD starts with a (column) orthonormal matrix $V_{s-1} \in \mathbb{R}^{d \times (s-1)}$. It finds an approximate leading eigenvector $v_s$ of $M_{s-1} = (I - V_{s-1} V_{s-1}^T) A A^T (I - V_{s-1} V_{s-1}^T)$, where $M_{s-1}$ is the projection of $M = AA^T$ into space $V_{s-1}^\perp$. Then, LazySVD appends $V_s \leftarrow [V_{s-1}, v_s]$ and continues to the next iteration.

Obtain Faster Running Time. Ideally, if each $v_s$ were exactly the leading eigenvector of $M_s$, then our final $V_k$ would become exactly the top $k$ singular vectors of $A$. However, computing exact eigenvectors is too slow, so the main challenge is to tolerate as much error as possible to compute each $v_s$, in order to tradeoff for a faster running time.

It was previously folklore that one can approximately compute each $v_s$ to a good precision so that the running time depends on all intermediate gaps $\frac{\sigma_i}{\sigma_i - \sigma_{i+1}}$ for $i = 1, \ldots, k$. This is too slow, although thought to be somewhat necessary by some authors. A weaker alternative is to compute each $v_s$ to an additive precision so $v_s^T M_{s-1} v_s \geq \sigma_s^2 - \text{gap} : \sigma_s^2$. However, this remains too slow because the running time would polynomially depend on $\sigma_1/\sigma_k$.

In LazySVD we tolerate the $s$-th leading eigenvalue computation to suffer from a multiplicative error $\text{gap}$ —more precisely, to satisfy $v_s^T M_{s-1} v_s \geq (1 - \text{gap})\|M_{s-1}\|_2$. This implies our declared running time in Table 1 owing to the 1-PCA result of Section 3. What it remains is to prove the correctness of our algorithm: that is, to prove $\|M_{s-1}\|_2 \approx \sigma_s^2$ for each $s = 1, 2, \ldots, k$.

Obtain Correctness. Our main idea is to use the fact that each vector $v_s$ “approximately” lies in the span of the top $k$ eigenvectors of $M = AA^T$.

Notice if each $v_s$ perfectly lay in the span of the top $k$ eigenvectors of $M$, we would be able to claim —using the Cauchy interlacing theorem— that (1) the $(k + 1 - s)$-th largest eigenvalue
of matrix $M_s$ would be never be larger than $\sigma^2_{k+1}$, and (2) the largest eigenvalue of $M_s$ would be never smaller than $\sigma^2_{s+1}$:

**Proposition 4.6** (Cauchy interlacing theorem). Given a symmetric matrix $N \in \mathbb{R}^{d \times d}$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$, if $v$ is in the span of the first $k$ eigenvectors of $N$, then $(I - vv^\top)N(I - vv^\top)$ has eigenvalues $\lambda'_1 \geq \cdots \geq \lambda'_{d-1}$ satisfying:

$$\forall i \in [k-1]: \lambda_i \geq \lambda'_i \geq \lambda_{i+1} \quad \text{and} \quad \forall i \geq k: \lambda_{i+1} = \lambda'_i.$$ 

Therefore, the difference between the largest and the $(k + 1 - s)$-th largest eigenvalue of $M_s$ would be at least $\sigma^2_{s+1} - \sigma^2_{k+1} > \Omega(\text{gap}) \cdot \sigma^2_{s+1}$ as long as $s = 0, 1, \ldots, k - 1$. This could allow us to apply a “gap-multiplicative error” algorithm to obtain the next leading eigenvector $v_{s+1}$ of $M_s$, and this $v_{s+1}$ would “almost completely” lie in the span of the top $(k - s)$ eigenvectors of $M_s$ and thus the span of the top $k$ eigenvectors of $M$. Repeating this argument for $k$ times, we could have obtained the top $k$ singular vectors of $A$, up to rotation.

Our main technique contribution is to extend the above argument into an approximate setting, and to propagate error “moderately” across iterations. While a naive bound could easily blow up the error exponentially with respect to $k$, we provide non-trivial analysis to show that the error grows at most linearly in $k$. This step of our proof essentially consists of two parts.

**Part 1:** We show that, in each iteration, the vector $v_s$ obtained from AppxPCA only correlates with the last $d - k$ eigenvectors of $M_{s-1}$ by a polynomially small factor (i.e., by $\text{poly}(\epsilon, 1/n, 1/d)$).

**Part 2:** We develop a gap-free variant of the Wedin theorem for matrices [23], which translates Part 1 into two statements.

- The last $d - k$ eigenvectors of $M_s = (I - vv^\top)M_{s-1}(I - vv^\top)$ approximately lie in the span of the last $d - k$ eigenvectors of $M_{s-1}$.
- The $(k + 1 - s)$-th eigenvalue of $M_s$ is close to the $(k + 2 - s)$-th eigenvalue of $M_{s-1}$.

Recursively applying the above statements $s$ times, we conclude that

- The last $d - k$ eigenvectors of $M_s$ approximately lie in the span of the last $d - k$ eigenvectors of $M_0 = M$.
- The $(k + 1 - s)$-th eigenvalue of $M_s$ is close to $\sigma^2_{k+1}$, the $(k + 1)$-th eigenvalue of $M_0 = M$.

These two properties replace the use of the Cauchy interlacing theorem, which only holds when “$v_s$ perfectly lay in the span of the top $k$ eigenvectors of $M$.” They together imply that “the difference between the largest and the $(k + 1 - s)$-th largest eigenvalue of $M_s$” is at least $\geq \sigma^2_{s+1} - \sigma^2_{k+1} > \Omega(\text{gap}) \cdot \sigma^2_{s+1}$. Therefore, we can proceed to the next iteration $s + 1$, and compute a gap-multiplicative approximate leading eigenvector $v_{s+1}$ of $M_s$.

This summarizes our intuition behind the correctness of LazySVD.

## 5 NNZ Running Time

In this section, we translate our results in the previous section into the $O(\text{nnz}(A) + \text{poly}(k, 1/\epsilon)(n + d))$ running-time statements. The idea is surprisingly simple: we sample either random columns of $A$, or random entries of $A$, and then apply LazySVD to compute the $k$-SVD. Such translation directly gives either $1/\epsilon^{2.5}$ results if AGD is used as the convex subroutine and either column or entry sampling is used, or a $1/\epsilon^{2}$ result if accelerated SVRG and column sampling are used together.

We only informally state our theorem and defer all the details to Appendix D.
**Theorem 5.1** (informal). Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$. For every $\varepsilon \in (0, 1/2)$, one can apply LazySVD with appropriately chosen $\delta_x$ on a “carefully sub-sampled version” of $A$. Then, the resulting matrix $V \in \mathbb{R}^{d \times k}$ can satisfy

- **spectral norm guarantee**: $\|A - V V^\top A\|_2 \leq \|A - A_k^*\|_2 + \varepsilon \|A - A_k^*\|_F$.

- **Frobenius norm guarantee**: $\|A - V V^\top A\|_F \leq (1 + \varepsilon) \|A - A_k^*\|_F$.

The total running time depends on (1) whether column or entry sampling is used, (2) which matrix inversion routine $A$ is used, and (3) whether spectral or Frobenius guarantee is needed. We list our deduced results in Table 2 and the formal statements can be found in Theorem D.4, D.6, and D.9.

**Remark 5.2.** The main purpose of our NNZ results is to demonstrate the strength of LazySVD framework in terms of improving the $\varepsilon$ dependency to $1/\varepsilon^2$. Since the $1/\varepsilon^2$ rate matches sampling complexity, it is very challenging have an NNZ result with $1/\varepsilon^2$ dependency. We have not tried hard, and believe it possible, to improve the polynomial dependence with respect to $k$ or $(\sigma_1/\sigma_{k+1})$. Also, somewhat surprisingly, in our analysis Frobenius norms become harder to minimize as opposed to spectral norms; this is in contrast to known literatures where usually Frobenius results are easier to prove [7].

6 Experiments

We demonstrate the practicality of our LazySVD framework, and compare it to block power method or block Krylov method. We emphasize that in theory, the best worse-case complexity for 1-PCA is obtained by AppxPCA on top of accelerated SVRG. However, for the size of our chosen datasets, Lanczos method runs faster than AppxPCA and therefore we adopt Lanczos method as the 1-PCA method for our LazySVD framework.

**Datasets.** We use datasets SNAP/amazon0302, SNAP/email-enron, and news20 that were also used by Musco and Musco [19], as well as an additional but famous dataset RCV1. The first two can be found on the SNAP website [16] and the last two can be found on the LibSVM website [11]. The four datasets give rise sparse matrices of dimensions $257570 \times 262111$, $35600 \times 16507$, $11269 \times 53975$, and $20242 \times 47236$ respectively.

**Implemented Algorithms.** For the block Krylov method, it is a well-known issue that the Lanczos type of three-term recurrence update is numerically unstable. This is why Musco and Musco [19] only used the stable variant of block Krylov which requires an orthogonalization of each $n \times k$ matrix with respect to all previously obtained $n \times k$ matrices. This greatly improves the numerical stability albeit sacrificing running time. We implement both these algorithms. In sum, we have implemented:

- **PM**: block power method for $T$ iterations.
- **Krylov**: stable block Krylov method for $T$ iterations [19].

---

4This is the best known spectral guarantee one can obtain using NNZ running time [7]. It is an open question whether the stricter $\|A - V V^\top A\|_2 \leq (1 + \varepsilon) \|A - A_k^*\|_2$ type of spectral guarantee is possible.

5On one hand, one can use dimension reduction such as [9] to reduce the problem size to $O(k/\varepsilon^3)$; to the best of our knowledge, it is impossible to obtain any NNZ result faster than $1/\varepsilon^3$ using solely dimension reduction. On the other hand, obtaining $1/\varepsilon^2$ dependency was the main contribution of [7]; they relied on alternating minimization but we have avoided it in our paper.

6Our LazySVD framework turns every 1-PCA method satisfying Theorem 3.1 (including Lanczos method) into a $k$-SVD solver. However, our theoretical results (esp. stochastic and NNZ) rely on AppxPCA because Lanczos is not a stochastic method.
We make the following observations:

- **Krylov(unstable)**: the three-term recurrence implementation of block Krylov for \( T \) iterations.
- **LazySVD**: \( k \) calls of the vanilla Lanczos method, and each call runs \( T \) iterations.

**A Fair Running-Time Comparison.** For a fixed integer \( T \), the four methods go through the dataset (in terms of multiplying \( A \) with column vectors) the same number of times. However, since LazySVD does not need block orthogonalization (as needed in PM and Krylov) and does not need a \((Tk)\)-dimensional SVD computation in the end (as needed in Krylov), the running time of LazySVD is clearly much faster for a fixed value \( T \). We therefore compare the performances of the four methods in terms of running time rather than \( T \).

We programmed the four algorithms using the same programming language with the same sparse-matrix implementation. We tested them single-threaded on the same Intel i7-3770 3.40GHz personal computer. As for the final low-dimensional SVD decomposition step at the end of the PM or Krylov method (which is not needed for our LazySVD), we used a third-party library that is built upon the x64 Intel Math Kernel Library so the time needed for such SVD is maximally reduced.

**Performance Metrics.** We compute four metrics on the output \( V = (v_1, \ldots, v_k) \in \mathbb{R}^{n \times k}:\)

- **Fnorm**: relative Frobenius norm error: 
  \[
  \frac{\|A - VV^T A\|_F - \|A - A_k^*\|_F}{\|A - A_k^*\|_F}.
  \]
- **spectral**: relative spectral norm error: 
  \[
  \frac{\|A - VV^T A\|_2 - \|A - A_k^*\|_2}{\|A - A_k^*\|_2}.
  \]
- **rayleigh(last)**: Rayleigh quotient error relative to \( \sigma_{k+1} \): 
  \[
  \max_{j=1}^k \frac{|\sigma_j^2 - v_j^T AA^T v_j|}{\sigma_{k+1}^2}.
  \]
- **rayleigh**: relative Rayleigh quotient error: 
  \[
  \max_{j=1}^k \frac{|\sigma_j^2 - v_j^T AA^T v_j|}{\sigma_j^2}.
  \]

The first three metrics were also used by Musco and Musco [19]. We added the fourth one because our theory only predicted convergence with respect to the fourth but not the third metric. However, we observe that in practice they are not much different from each other.

**Our Results.** We study four datasets each with \( k = 10, 20, 30 \) and with the four performance metrics, totaling 48 plots. Due to space limitation, we only select six representative plots out of 48 and include them in Figure 1. (The full plots can be found in Figure 2, 3, 4 and 5 in the appendix.) We make the following observations:

- **LazySVD** outperforms its three competitors almost universally.
- **Krylov(unstable)** outperforms Krylov for small value \( T \); however, it is less useful for obtaining accurate solutions due to its instability. (The dotted green curves even go up if \( T \) is large.)
• Subspace power method performs the slowest unsurprisingly due to its lack of acceleration.

Acknowledgements

We thank Ohad Shamir for listening to our ideas and pointing us to some of the existing results. The first author is partially supported by a Microsoft Research Award, no. 0518584, and an NSF grant, no. CCF-1412958.

APPENDIX

A Proof Details for Theorem 3.1: Convergence of AppxPCA

A.1 Inexact Power Method

Consider the classical power method that is to start with a random unit vector $w_0$ and apply $w_t \leftarrow Mw_{t-1}/\|Mw_{t-1}\|$ iteratively.

Lemma A.1 (Exact Power Method). Let $M$ be a PSD matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ and the corresponding eigenvectors $u_1, \ldots, u_d$. Fix an error tolerance $\varepsilon > 0$, parameter $\kappa \geq 1$, and failure probability $p > 0$, define

$$T_{PM}(\kappa, \varepsilon, p) = \left\lceil \frac{\kappa}{2} \log \left( \frac{9d}{p^2 \varepsilon} \right) \right\rceil$$

Then, with probability at least $1 - p$ it holds that $\forall t \geq T_{PM}(\kappa, \varepsilon, p)$:

$$\sum_{i \in [d], \lambda_i \leq (1 - 1/\kappa) \lambda_1} (w_t^\top u_i)^2 \leq \varepsilon \quad \text{and} \quad w_t^\top Mw_t \geq (1 - 1/\kappa - \varepsilon) \lambda_1 .$$

The probability of success depends only on the random variable $(w_0^\top u_1)^2$.

Proof. For each $i \in [d]$,

$$\frac{(w_t^\top u_i)^2}{\lambda_i} \leq \frac{(w_0^\top u_i)^2}{\lambda_i} \left( \frac{\lambda_1}{\lambda_i} \right)^{2t} = \frac{(w_0^\top u_i)^2}{\lambda_i} \left( \frac{\lambda_1}{\lambda_i} \right)^{2t}$$

Since $w_0$ is a random unit vector, according to for instance Lemma 5 of [6], it holds with probability at least $1 - p$ that $(w_0^\top u_1)^2 \geq \frac{p^2}{9d}$. Substituting this into the above inequality, we conclude that with probability at least $1 - p$, for all $i \in [d]$, we have

$$(w_0^\top u_i)^2 \leq (w_0^\top u_i) \cdot \frac{9d}{p^2} \left( 1 - \frac{\lambda_1 - \lambda_i}{\lambda_1} \right)^{2t} \leq (w_0^\top u_i) \cdot \frac{9d}{p^2} \cdot \exp \left( - \frac{2 \lambda_1 - \lambda_i}{\lambda_1} t \right)$$

As a result, for every $t \geq T_{PM}(\kappa, \varepsilon, p)$, and every $i$ such that $\lambda_i \leq (1 - 1/\kappa) \lambda_1$ (which implies $\frac{\lambda_1 - \lambda_i}{\lambda_1} \geq 1/\kappa$), we have

$$(w_t^\top u_i)^2 \leq \varepsilon \cdot (w_0^\top u_i)^2$$
Summing them up we have
\[
\sum_{i \in [d], \lambda_i \leq (1 - 1/\kappa)\lambda_1} (w_i^\top u_i)^2 \leq \varepsilon \sum_{i \in [d]} (w_i^0 u_i)^2 = \varepsilon .
\]

This finishes the proof of the first bound. To prove the second bound, we compute that
\[
w_i^\top M w_t = \sum_{i=1}^d \lambda_i (w_i^\top u_i)^2 \geq \sum_{i \in [d], \lambda_i > (1 - 1/\kappa)\lambda_1} \lambda_i (w_i^\top u_i)^2 \geq (1 - 1/\kappa)\lambda_1 \cdot \sum_{i \in [d], \lambda_i > (1 - 1/\kappa)\lambda_1} (w_i^\top u_i)^2
\]
\[
\geq (1 - 1/\kappa)(1 - \varepsilon)\lambda_1 \geq (1 - 1/\kappa - \varepsilon)\lambda_1 .
\]

□

**Lemma A.2** (Lemma 4.1 of [12]). Let \( M \) be a PSD matrix with eigenvalues \( \lambda_1 \geq \cdots \geq \lambda_d \). Fix an accuracy parameter \( \varepsilon > 0 \), and consider two update sequences

\[
\hat{w}_0^* = w_0, \quad \forall t \geq 1: \hat{w}_t^* \leftarrow M \hat{w}_{t-1}^*
\]
\[
\hat{w}_0 = w_0, \quad \forall t \geq 1: \hat{w}_t \text{ satisfies } \|\hat{w}_t - M \hat{w}_{t-1}\| \leq \varepsilon ,
\]

Then, defining \( w_t = \hat{w}_t/\|\hat{w}_t\| \) and \( w_t^* = \hat{w}_t^*/\|\hat{w}_t^*\| \), it satisfies

\[
\|w_t - w_t^*\| \leq \varepsilon \cdot \Gamma(M, t),
\]

where

\[
\Gamma(M, t) \overset{\text{def}}{=} \frac{2}{\lambda_d} \left\{ t, \frac{(\lambda_1^t - 1)}{\lambda_1 - 1}, \frac{\lambda_1^t}{\lambda_1} \right\} \text{ if } \lambda_1 \geq 1; \quad \text{and we have } \Gamma(M, t) \leq 2t \cdot \max\{1, \lambda_1^t\} \lambda_1^t.
\]

**Theorem A.3** (Inexact Power Method). Let \( M \) be a PSD matrix with eigenvalues \( \lambda_1 \geq \cdots \geq \lambda_d \) and the corresponding eigenvectors \( u_1, \ldots, u_d \). With probability at least \( 1 - p \) it holds that, for every \( \varepsilon \in (0, 1) \) and every \( t \geq T_{\text{PM}}(\kappa, \varepsilon/4, p) \), if \( w_t \) is generated by the power method with per-iteration error \( \varepsilon = \frac{\varepsilon}{4\Gamma(M, t)} \), then

\[
\sum_{i \in [d], \lambda_i \leq (1 - 1/\kappa)\lambda_1} (w_i^t u_i)^2 \leq \varepsilon \quad \text{and} \quad w_i^\top M w_t \geq (1 - 1/\kappa - \varepsilon)\lambda_1 .
\]

**Proof.** Denoting by \( w_t^* \) the output of power method with exact updates (with the same starting vector \( w_0 \) following Lemma A.2)

\[
\sum_{j \in [d], \lambda_j \leq (1 - 1/\kappa)\lambda_1} (w_j^t u_j)^2 = \sum_{j \in [d], \lambda_j \leq (1 - 1/\kappa)\lambda_1} (\langle w_j^*, u_j \rangle + \langle w_t - w_j^*, u_j \rangle)^2
\]
\[
\leq \sum_{j \in [d], \lambda_j \leq (1 - 1/\kappa)\lambda_1} 2(\langle w_j^*, u_j \rangle)^2 + 2(\langle w_t - w_j^*, u_j \rangle)^2
\]
\[
\leq \frac{\varepsilon}{2} + 2 \sum_{j \in [d]} (w_t - w_j^*)^\top u_j u_j^\top (w_t - w_j^*) = \frac{\varepsilon}{2} + 2\|w_t - w_j^*\|^2 \leq \varepsilon .
\]

Above, the first inequality is because \((a + b)^2 \leq 2a^2 + 2b^2\), the second inequality is due to the definition of \( w_t^* \) and Lemma A.1, and the last inequality is because \( \|w_t - w_j^*\| \leq \varepsilon \cdot \Gamma(M, t) = \frac{\varepsilon}{4} \) which implies \( 2\|w_t - w_j^*\|^2 \leq \varepsilon^2/8 < \varepsilon/2 \).

This finishes the proof of the first bound. The proof of the second bound is identical to the last paragraph of the proof of Lemma A.1. □
A.2 Proof of Theorem 3.1

Lemma A.4. With probability at least $1 - p$, it holds that (where $\lambda_1$ is the largest eigenvalue of $M$.)

(a) $\tilde{\varepsilon}_1 \leq \frac{1}{32(\lambda_1 - \lambda_1 \mathbf{I} - M)^{-1} m_1}$ for each iteration $s$;

(b) $\tilde{\varepsilon}_2 \leq \frac{1}{4\lambda_1(M^{-1} - M)^{-1} M_2}$;

(c) $0 \leq \frac{1}{2}(\lambda(s) - \lambda_1) \leq \Delta(s) \leq \lambda(s) - \lambda_1$ and $\frac{1}{2}(\lambda(s) - \lambda_1) \leq \lambda(s) - \lambda_1$ for each iteration $s$;

(d) $\lambda(f) - \lambda_1 \in \left(\frac{\delta_x}{12} \lambda(f), \delta_x \lambda_1 \right]$ when the repeat-until loop is over.

Proof. We denote by $A(s) \triangleq (\lambda(s) I - M)^{-1}$ for notational simplicity. Below we prove all the items by induction for a specific iteration $s \geq 2$ assuming that the items of the previous $s - 1$ iterations are true. The base case of $s = 1$ can be verified similar to the general arguments below but requiring some non-trivial notational changes. We omitted the proof of the base case $s = 1$ in this paper.

(a) Recall that

$$
\Gamma(A(s-1), t) \leq 2t \cdot \frac{\max\{1, \lambda_{\max}(A(s-1))\}}{\lambda_{\min}(A(s-1))}
$$

On one hand, we have $\lambda_{\max}(A(s-1)) = \frac{\lambda(s-1)}{\lambda(s-1) - \lambda_1} \leq \frac{2}{\lambda(s-2) - \lambda_1} \leq \frac{2}{\Delta(s-1)}$ using [Lemma A.4.c] of the previous iteration. Combining this with the termination criterion $\Delta(s) \geq \frac{1}{\lambda(s-1) - \lambda_1}$, we have $\lambda_{\max}(A(s-1)) \leq \frac{6}{\delta_x \lambda(s-1)}$. On the other hand, we have $\lambda_{\min}(A(s-1)) = \frac{1}{\lambda(s-1) - \lambda_1} \geq \frac{1}{\lambda(s-1)}$. Combining the two bounds we conclude that $\Gamma(A(s-1), t) \leq 2t(6/\delta_x)^t$. It is now obvious that $\tilde{\varepsilon}_1 \leq \frac{1}{32(\lambda_1 - \lambda_1 \mathbf{I} - M)^{-1} m_1}$ is satisfied because $\tilde{\varepsilon}_1 = \frac{\delta_x}{6} \left(\frac{1}{5}\right)^{m_1}$.

(b) The same analysis as in the proof of Lemma A.4.a suggests that $\Gamma(f, t) \leq 2t(6/\delta_x)^t$. This immediately yields $\tilde{\varepsilon}_2 \leq \frac{1}{4\lambda_1(M^{-1} - M)^{-1} M_2}$ because $\tilde{\varepsilon}_2 = \frac{\delta_x}{8} \left(\frac{1}{5}\right)^{m_2}$.

(c) Because Lemma A.4.a holds for the current iteration $s$ we can apply Theorem A.3 (with $\varepsilon = \frac{1}{8}$ and $\kappa = 8$) and get

$$
w^\top A(s-1) w \geq \frac{3}{4} \lambda_{\max}(A(s-1)).
$$

By the definition of $v$ in AppxPCA and the Cauchy-Schwartz inequality it holds that

$$
w^\top v = w^\top A(s-1) w + w^\top (v - A(s-1) w) \in \left[w^\top A(s-1) w - \tilde{\varepsilon}_1, w^\top A(s-1) w + \tilde{\varepsilon}_1 \right]
$$

Combining the above two equations we have

$$
w^\top v - \tilde{\varepsilon}_1 \in \left[\frac{3}{4} \lambda_{\max}(A(s-1)) - 2\varepsilon_1, \lambda_{\max}(A(s-1)) \right] \subseteq \left[\frac{1}{2} \lambda_{\max}(A(s-1)), \lambda_{\max}(A(s-1)) \right] = \left[\frac{1}{2}, 1\right] \cdot \frac{1}{\lambda(s-1) - \lambda_1}.
$$

In other words, $\Delta(s) \triangleq \frac{1}{2} \cdot \frac{1}{w^\top v - \tilde{\varepsilon}_1} \in \left[\frac{1}{2} \lambda(s-1) - \lambda_1, \lambda(s-1) - \lambda_1\right]$.

At the same time, our update rule $\lambda(s) = \lambda(s-1) - \Delta(s) / 2$ ensures that $\lambda(s) - \lambda_1 = \lambda(s-1) - \lambda_1 - \Delta(s) / 2 \geq \lambda(s-1) - \lambda_1 - \Delta(s) / 2 = \frac{1}{2} \lambda(s-1) - \lambda_1$. 

(d) The upper bound holds because \( \lambda(f) - \lambda_1 = \lambda(f-1) - \Delta(f) \leq \frac{3}{2} \Delta(f) \leq \frac{\delta_x \lambda(f)}{2} \) where the first inequality follows from Lemma A.4.c of this last iteration, and the second inequality follows from our termination criterion \( \Delta(f) \leq \frac{\delta_x \lambda(f)}{2} \). Simply rewriting this inequality we have \( \lambda(f) - \lambda_1 \leq \frac{\delta_x / 2}{1 - \delta_x / 2} \lambda_1 \leq \delta_x \lambda_1 \).

The lower bound is because using Lemma A.4.c (of this and the previous iteration) we have \( \lambda(f) - \lambda_1 \geq \frac{1}{4} (\lambda(f-2) - \lambda_1) \geq \frac{\Delta(f-2)}{4} \geq \frac{\delta_x \lambda(f-2)}{12} \geq \frac{\delta_x \lambda(f)}{12} \). Here, inequality \( \Phi \) is because \( \Delta(f-2) > \frac{\delta_x \lambda(f-1)}{3} \) due to the termination criterion.

Finally since the success of Theorem A.3 only depends on the randomness of \( \hat{w}_0 \), we have that with probability at least \( 1 - p \) that all the above items are satisfied.

**Proof of Theorem 3.1.** It follows from Theorem A.3 (with \( \kappa = 2 \)) that, letting \( \mu_i = 1/(\lambda(f) - \lambda_i) \) be the \( i \)-th largest eigenvalue of the matrix \( (\lambda(f)I - M)^{-1} \), then

\[
\sum_{i \in [d], \mu_i \leq \mu_i / 2} (w^\top u_i)^2 \leq \varepsilon
\]

Note that if an index \( i \in [d] \) satisfies \( \lambda_1 - \lambda_i \geq \delta_x \lambda_1 \), then we must have \( \lambda_1 - \lambda_i \geq \lambda(f) - \lambda_1 \) owing to \( \lambda(f) - \lambda_1 \leq \delta_x \lambda_1 \) from Lemma A.4.d. This further implies that \( 2 \lambda(f) - \lambda_1 \leq \lambda(f) - \lambda_i \) and therefore \( \mu_i / 2 \geq \mu_i \). In sum, we also have

\[
\sum_{i \in [d], \lambda_i \leq (1 - \delta_x) \lambda_1} (w^\top u_i)^2 \leq \varepsilon.
\]

On the other hand,

\[
w^\top Mw = \sum_{i=1}^{d} \lambda_i (w^\top u_i)^2 \geq \sum_{i \in [d], \lambda_i > (1 - \delta_x) \lambda_1} \lambda_i (w^\top u_i)^2 \geq (1 - \delta_x) \lambda_1 \cdot \sum_{i \in [d], \lambda_i > (1 - \delta_x) \lambda_1} (w^\top u_i)^2 \\
\geq (1 - \delta_x)(1 - \varepsilon) \lambda_1.
\]

The number of oracle calls to \( A \) is determined by the number of iterations in the repeat-until loop. It is easy to verify that there are at most \( O(\log(1/\delta_x)) \) such iterations, so the total number of oracle calls to \( A \) is only \( O(\log(1/\delta_x) m_1 + m_2) \).

In addition, each time we call \( A \) we have

\[
\frac{\lambda(s)}{\lambda_{\min}(\lambda(s)I - M)} \leq \frac{\lambda(s)}{\lambda(s) - \lambda_1}
\]

If \( s = 0 \) then we have \( \frac{\lambda^{(0)}}{\delta_x \lambda_1} \leq \frac{1 + \delta_x}{\delta_x} \) because \( \lambda_1 \leq 1 \). If \( s \leq f - 2 \) then we have \( \frac{\lambda_{\min}(\lambda(s)I - M)}{\lambda(s) - \lambda_1} \leq \frac{\lambda(s)}{\Delta(s + 1)} \leq \frac{\lambda(s)}{\Delta(s + 1)} \leq \frac{2 \lambda(s)}{\delta_x \lambda(s) / 3} = \frac{6}{\delta_x} \) where the first inequality follows from Lemma A.4.c, the second one follows from the stopping criterion, and the last one follows from the monotonicity of \( \lambda(s) \). If \( s = f - 1 \) then we have \( \frac{\lambda_{\min}(\lambda(s)I - M)}{\lambda(s) - \lambda_1} \leq \frac{2 \lambda(s)}{\delta_x \lambda(s) / 3} = \frac{6}{\delta_x} \) where the first two inequalities follow from Lemma A.4.c and the third inequality follows from our stopping criterion. If \( s = f \) then we have \( \frac{\lambda_{\min}(\lambda(s)I - M)}{\lambda(s) - \lambda_1} \leq \frac{12}{\delta_x} \) owing to [Lemma A.4.d]. In all cases we have \( \frac{\lambda(s)}{\lambda_{\min}(\lambda(s)I - M)} \leq \frac{12}{\delta_x} \) where the last inequality follows from \( \lambda(s) \geq \lambda_1 \).
B Lemmas Needed for Proving Our Main Theorem

In this section we provide some necessary lemmas on matrices that shall become essential for our proof of Theorem 4.1.

**Proposition B.1.** Let $A, B$ be two (column) orthonormal matrix such that for $\eta \geq 0$,
\[
A^TBB^TA \succeq (1 - \eta)I
\]

Then we have: there exists a matrix $Q, \|Q\|_2 \leq 1$ such that
\[
\|A - BQ\|_2 \leq \sqrt{\eta}
\]

**Proof.** Since $A^TA = I$ and $A^TBB^TA \succeq (1 - \eta)I$, we know that $A^TB(B^\perp)^TA \leq \eta I$. By the fact that
\[
A = (BB^T + B^\perp(B^\perp)^T)A = BB^TA + B^\perp(B^\perp)^TA
\]
we can let $Q = B^TA$ and obtain
\[
\|A - BQ\|_2 \leq \|B^\perp(B^\perp)^TA\|_2 \leq \sqrt{\eta} .
\]

\[ \square \]

B.1 Approximate Projection Lemma

**Lemma B.2.** Let $M$ be a PSD matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ and the corresponding eigenvectors $u_1, \ldots, u_d \in \mathbb{R}^d$. For every $k \geq 1$, define $U^\perp = (u_1, \ldots, u_k) \in \mathbb{R}^{d \times k}$ and $U = (u_{k+1}, \ldots, u_d) \in \mathbb{R}^{d \times (d-k)}$. For every $\varepsilon \in (0, \frac{1}{2})$, let $V_s \in \mathbb{R}^{d \times s}$ be an orthogonal matrix such that $\|V_s^TU\|_2 \leq \varepsilon$, define $Q_s \in \mathbb{R}^{d \times s}$ to be an arbitrary orthogonal basis of the column span of $U^\perp(U^\perp)^TV_s$, then we have:
\[
\left\|\left( I - Q_sQ_s^T \right)M\left( I - Q_sQ_s^T \right) - \left( I - V_sV_s^T \right)M\left( I - V_sV_s^T \right) \right\|_2 \leq 13\lambda_1\varepsilon .
\]

**Proof of Lemma B.2.** Since $Q_s$ is an orthogonal basis of the column span of $U^\perp(U^\perp)^TV_s$, there is a matrix $R \in \mathbb{R}^{s \times s}$ such that
\[
Q_s = U^\perp(U^\perp)^TV_sR
\]

Using the fact that $Q_s^TQ_s = I$, we have:
\[
(U^\perp(U^\perp)^TV_sR)^T(U^\perp(U^\perp)^TV_sR) = I \implies R^TV_s^TU^\perp(U^\perp)^TV_sR = I .
\]

By the fact that $V_s^TV_s = I$ and $U^\perp(U^\perp)^T + UU^T = I$, we can rewrite the above equality as:
\[
R^T\left( I - V_s^TUU^TV_s \right)R = I \quad (B.1)
\]

From our lemma assumption, we have: $\|V_s^TU\|_2 \leq \varepsilon$, which implies $0 \preceq V_s^TUU^TV_s \preceq \varepsilon^2 I$. Putting this into $(B.1)$, we obtain:
\[
I \preceq R^TR \preceq \frac{1}{1 - \varepsilon^2} I \preceq (1 + \frac{4}{3}\varepsilon^2)I
\]
The above inequality directly implies that \( I \preceq RR^T \preceq (1 + \frac{4}{3} \varepsilon^2)I \). Therefore,

\[
\left\| Q_sQ_s^T - V_sV_s^T \right\|_2^2 = \left\| U^\perp(U^\perp)^T V_s R R^T V_s^T U^\perp(U^\perp)^T - V_sV_s^T \right\|_2^2 \\
= \left\| U^\perp(U^\perp)^T V_s R R^T V_s^T U^\perp(U^\perp)^T - U^\perp(U^\perp)^T + UU^T \right\|_2^2 \\
\leq \left\| U^\perp(U^\perp)^T V_s (R R^T - I) V_s^T U^\perp(U^\perp)^T \right\|_2^2 + \left\| UU^TV_sV_s^TUU^T \right\|_2^2 + 2 \left\| U^\perp(U^\perp)^T V_s V_s^T UU^T \right\|_2^2 \\
\leq \left\| RR^T - I \right\|_2 + \left\| U^T V_s^T U \right\|_2 + 2 \left\| V_s^T UU^T V_s \right\|^{1/2} \\
\leq \frac{4}{3} \varepsilon^2 + \varepsilon^2 + 2\varepsilon < \frac{19}{6} \varepsilon .
\]

Finally, we have

\[
\left\| (I - Q_sQ_s^T) M (I - Q_sQ_s^T) - (I - V_sV_s^T) M (I - V_sV_s^T) \right\|_2^2 \\
\leq 2 \left\| (Q_sQ_s^T - V_sV_s^T) M \right\|_2^2 + \left\| (Q_sQ_s^T - V_sV_s^T) MQ_sQ_s^T \right\|_2^2 + \left\| (Q_sQ_s^T - V_sV_s^T) MV_sV_s^T \right\|_2^2 \\
\leq \frac{19 \times 4}{6} \lambda_1 \varepsilon < 13 \lambda_1 \varepsilon .
\]

\( \square \)

### B.2 Gap-Free Wedin Theorem

**Lemma B.3 (Gap free Wedin Theorem).** For \( \varepsilon \geq 0 \), let \( A, B \) be two PSD matrices such that \( \| A - B \|_2 \leq \varepsilon \). For every \( \mu \geq 0 \), \( \tau > 0 \), let \( U \) be column orthonormal matrix consisting of eigenvectors of \( A \) with eigenvalue \( \leq \mu \), let \( V \) be column orthonormal matrix consisting of eigenvectors of \( B \) with eigenvalue \( \geq \mu + \tau \), then we have:

\[ \| U^T V \| \leq \frac{\varepsilon}{\tau} . \]

**Proof of Lemma B.3.** We write \( A \) and \( B \) in terms of eigenvalue decomposition:

\[ A = U \Sigma U^T + U' \Sigma' U'^T \quad \text{and} \quad B = V \tilde{\Sigma} V^T + V' \tilde{\Sigma}' V'^T , \]

where \( U' \) is orthogonal to \( U \) and \( V' \) is orthogonal to \( V \). Letting \( R = A - B \), we obtain:

\[ \Sigma U^T = U^T A = U^T (B + R) \]

\[ \implies \Sigma U^T V = U^T BV + U^T RV = U^T V \tilde{\Sigma} + U^T RV \]

\[ \implies \Sigma U^T V \tilde{\Sigma}^{-1} = U^T V + U^T RV \tilde{\Sigma}^{-1} . \]

Taking spectral norm on both sides, we obtain:

\[ \| \Sigma \|_2 \| U^T V \|_2 \| \tilde{\Sigma}^{-1} \|_2 \geq \| \Sigma U^T V \tilde{\Sigma}^{-1} \|_2 \geq \| U^T V \|_2 - \| U^T RV \tilde{\Sigma}^{-1} \|_2 . \]

This can be simplified to

\[ \frac{\mu}{\mu + \tau} \| U^T V \|_2 \geq \| U^T V \|_2 - \frac{\varepsilon}{\mu + \tau} , \]

and therefore we have \( \| U^T V \|_2 \leq \frac{\varepsilon}{\tau} \) as desired. \( \square \)
B.3 Projected Power Method

Lemma B.4. Let \( M \in \mathbb{R}^{d \times d} \) be a PSD matrix with eigenvalues \( \lambda_1 \geq \cdots \geq \lambda_d \) and corresponding eigenvectors \( u_1, \ldots, u_d \). Define \( U = (u_{j+1}, \ldots, u_d) \in \mathbb{R}^{d \times (d-j)} \) to be the matrix consisting of all eigenvectors with eigenvalue \( \leq \mu \). Let \( v \in \mathbb{R}^d \) be a unit vector such that \( \|v^\top U\|_2 \leq \varepsilon \leq 1/2 \), and define

\[
M' = \left( I - vv^\top \right) M \left( I - vv^\top \right)
\]

Then, denoting \( [V_2, V_1, v] \in \mathbb{R}^{d \times d} \) as the unitary matrix consisting of (column) eigenvectors of \( M' \) with descending eigenvalues, where \( V_1 \) consists of eigenvectors with eigenvalue \( \leq \mu + \tau \), then there exists a matrix \( Q, \|Q\|_2 \leq 1 \) such that

\[
\|U - V_1 Q\|_2 \leq \sqrt{\frac{169\lambda_1^2 \varepsilon^2}{\tau^2} + \varepsilon^2}
\]

Proof of Lemma B.4. Using Lemma B.2, let \( q = \frac{U^+ (U^+)^\top v}{\|U^+(U^+)^\top v\|_2} \) be the projection of \( v \) to \( U^\perp \), we know that

\[
\left\| \left( I - qq^\top \right) M \left( I - qq^\top \right) - \left( I - vv^\top \right) M \left( I - vv^\top \right) \right\|_2 \leq 13\lambda_1 \varepsilon
\]

Denote \( (I - qq^\top) M (I - qq^\top) \) as \( M'' \). We know that \( u_{j+1}, \ldots, u_d \) are still eigenvectors of \( M'' \) with eigenvalue \( \lambda_{j+1}, \ldots, \lambda_d \).

Apply Lemma B.3 on \( A = M'' \), \( U \) and \( B = M', \ V = V_2 \), we obtain:

\[
\|U^\top V_2\|_2 \leq \frac{13\lambda_1 \varepsilon}{\tau}
\]

This implies that

\[
U^\top V_1 V_1^\top U = I - U^\top V_2 V_2^\top U - U^\top vv^\top U \succeq \left( 1 - \frac{169\lambda_1^2 \varepsilon^2}{\tau^2} - \varepsilon^2 \right) I
\]

where the inequality uses the assumption \( \|v^\top U\|_2 \leq \varepsilon \).

Apply Proposition B.1 to \( A = U \) and \( B = V_1 \), we conclude that there exists a matrix \( Q, \|Q\|_2 \leq 1 \) such that

\[
\|U - V_1 Q\|_2 \leq \sqrt{\frac{169\lambda_1^2 \varepsilon^2}{\tau^2} + \varepsilon^2}
\]

\[\Box\]

C Proof Details for Theorem 4.1: Our Main Theorem

In this section we prove Theorem 4.1 formally.
Theorem 4.1 (restated). Let $M \in \mathbb{R}^{d \times d}$ be a symmetric matrix with eigenvalues $1 \geq \lambda_1 \geq \cdots \lambda_d \geq 0$ and corresponding eigenvectors $u_1, \ldots, u_d$. Let $k \in [d]$, and $\delta_x, \varepsilon_{\text{pca}}, p \in (0, 1)$. Then, LazySVD outputs a (column) orthonormal matrix $V_k = (v_1, \ldots, v_k) \in \mathbb{R}^{d \times k}$ which, with probability at least $1 - p$, satisfies all of the following properties.

(Denote by $M_k = (I - V_k V_k^\top) M (I - V_k V_k^\top)$.)

(a) Core lemma: if $\varepsilon_{\text{pca}} \leq \frac{\delta_x^2}{2 \sqrt{k} (\lambda_1 / \lambda_k)^{1/2}}$, then $\|V_k^\top U\|_2 \leq \varepsilon$, where $U = (u_j, \ldots, u_d)$ is the (column) orthonormal matrix and $j$ is the smallest index satisfying $\lambda_j \leq (1 - \delta_x) \|M_{k-1}\|_2$.

(b) Spectral norm guarantee: if $\varepsilon_{\text{pca}} \leq \frac{\delta_x^2}{2 \sqrt{k} (\lambda_1 / \lambda_{k+1})^{3/2}}$, then $\lambda_{k+1} \leq \|M_k\|_2 \leq \frac{\lambda_{k+1}}{1 - \delta_x}$. 

(c) Rayleigh quotient guarantee: if $\varepsilon_{\text{pca}} \leq \frac{\delta_x^2}{2 \sqrt{k} (\lambda_1 / \lambda_{k+1})^{3/2}}$, then $(1 - \delta_x) \lambda_k \leq v_k^\top M v_k \leq \frac{1}{1 - \delta_x} \lambda_k$.

(d) Schatten-q norm guarantee: for every $q \geq 1$, if $\varepsilon_{\text{pca}} \leq \frac{\delta_x^2}{2 \sqrt{k} q (\lambda_1 / \lambda_{k+1})^{1/2}}$, then $\|M_k\|_{s_q} \leq \left(\frac{1 + \delta_x^2}{(1 - \delta_x)^2} \left(\sum_{i=k+1}^d \lambda_i^q\right)\right)^{1/q}$.

Proof of Theorem 4.1. Let $V_s = (v_1, \ldots, v_s)$, so we can write

$$M_s = (I - V_s V_s^\top) M (I - V_s V_s^\top) = (I - v_s v_s^\top) M_{s-1} (I - v_s v_s^\top)$$

(a) Define $\widehat{\lambda} = \|M_{k-1}\|_2 \geq \lambda_k$.

Note that all column vectors in $V_s$ are automatically eigenvectors of $M_s$ with eigenvalues zero. For analysis purpose only, let $W_s$ be the column matrix of eigenvectors in $V_s^\perp$ of $M_s$ that have eigenvalues in the range $[0, (1 - \delta_x + \tau_s)\widehat{\lambda}]$, where $\tau_s \overset{\text{def}}{=} \frac{2}{2k} \delta_x$. We now show that for every $s = 0, \ldots, k$, there exists a matrix $Q_s$ such that $\|U - W_s Q_s\|_2$ is small and $\|Q_s\|_2 \leq 1$. We will do this by induction.

In the base case: since $\tau_0 = 0$, we have $W_0 = U$ by the definition of $U$. We can therefore define $Q_0$ to be the identity matrix.

For every $s = 0, 1, \ldots, k - 1$, suppose there exists a matrix $Q_s$ with $\|Q_s\|_2 \leq 1$ that satisfies $\|U - W_s Q_s\|_2 \leq \eta_s$ for some $\eta_s > 0$, we construct $Q_{s+1}$ as follows.

First we observe that \textbf{ApexPCA} outputs a unit vector $\nu_{s+1}^\top$ satisfying $\|\nu_{s+1}^\top W_s\|_2 \leq \varepsilon_{\text{pca}}$ and $\|\nu_{s+1}^\top V_s\|_2 \leq \varepsilon_{\text{pca}}$ with probability at least $1 - p/k$. This follows from Theorem 3.1 because $[0, (1 - \delta_x + \tau_s)\widehat{\lambda}] \subseteq [0, (1 - \delta_x/2)\widehat{\lambda}]$, together with the fact that $\|M_s\|_2 \geq \|M_{k-1}\|_2 \geq \lambda$. Now, since $v_{s+1}$ is the projection of $\nu_{s+1}^\top$ into $V_{s}^\perp$, we have

$$\|v_{s+1}^\top W_s\|_2^2 \leq \frac{\|\nu_{s+1}^\top W_s\|_2^2}{\|I - V_s V_s^\top\|_2^2} = \frac{\|\nu_{s+1}^\top W_s\|_2^2}{1 - \|V_s^\top \nu_{s+1}\|_2^2} \leq \frac{\varepsilon_{\text{pca}}}{1 - \varepsilon_{\text{pca}}} < 1.5 \varepsilon_{\text{pca}} . \quad (C.1)$$

Next we apply \textbf{Lemma B.4} with $M = M_s$, $M' = M_{s+1}$, $U = W_s$, $V = W_{s+1}$, $v = v_{s+1}$, $\mu = (1 - \delta_x + \tau_s)\lambda$, and $\tau = (\tau_{s+1} - \tau_s)\lambda$. We obtain a matrix $\tilde{Q}_s$, $\|Q_s\|_2 \leq 1$ such that\footnote{Technically speaking, to apply \textbf{Lemma B.4} we need $U = W_s$ to consist of all eigenvectors of $M_s$ with eigenvalues $\leq \mu$. However, we only defined $W_s$ to be eigenvectors of $M_s$ with eigenvalues $\leq \mu$ that are also orthogonal to $V_s$. It is straightforward to verify that the same result of \textbf{Lemma B.4} remains true because $v_{s+1}$ is orthogonal to $V_s$.} 

$$\|W_s - W_{s+1} \tilde{Q}_s\|_2 \leq \sqrt{\frac{169 (\lambda_1 / \lambda) \cdot 1.5 \varepsilon_{\text{pca}}}{(\tau_{s+1} - \tau_s)^2} + \varepsilon_{\text{pca}}} < \frac{32 k \sqrt{\varepsilon_{\text{pca}}}}{\lambda \delta_x} .$$
and this implies that
\[
\|W_{s+1}Q_s - U\|_2 \leq \|W_{s+1}Q_s - W_sQ_s\|_2 + \|W_sQ_s - U\|_2 \leq \eta_s + \frac{32\lambda_k k^{\varepsilon_{\text{pca}}}}{\lambda_k \delta_x}.
\]

Let \(Q_{s+1} = \tilde{Q}_sQ_s\) we know that \(\|Q_{s+1}\|_2 \leq 1\) and
\[
\|W_{s+1}Q_{s+1} - U\|_2 \leq \eta_{s+1} \equiv \eta_s + \frac{32\lambda_1 k^{\varepsilon_{\text{pca}}}}{\lambda_k \delta_x}.
\]

Therefore, after \(k\)-iterations of LazySVD, we obtain:
\[
\|W_kQ_k - U\|_2 \leq \eta_k = \frac{32\lambda_1 k^2 \sqrt{\varepsilon_{\text{pca}}}}{\delta_x}.
\]

Multiply \(U^T\) from the left, we obtain \(\|U^TW_kQ_k - I\|_2 \leq \eta_k\). Since \(\|Q_k\|_2 \leq 1\), we must have \(\sigma_{\text{min}}(U^TW_k) \geq 1 - \eta_k\) (here \(\sigma_{\text{min}}\) denotes the smallest singular value). Therefore,
\[
U^TW_kW_k^TU \succeq (1 - \eta_k)^2 I.
\]

Since \(V_k\) and \(W_k\) are orthogonal of each other, we have
\[
U^TV_kV_k^TU \leq U^T(I - W_kW_k^T)U \leq I - (1 - \eta_k)^2 I \leq 2\eta_k I
\]

Therefore,
\[
\|V_k^TU\|_2 \leq 8\left(\frac{\lambda_1}{\lambda_k}\right)^{1/2}k^{1/4}\delta_{\text{pca}}^2 \leq \varepsilon.
\]

(b) The statement is obvious when \(k = 0\). For every \(k \geq 1\), the lower bound is obvious. We prove the upper bound by contradiction. Suppose that \(\|M_k\|_2 > \frac{\lambda_{k+1}}{1 - \delta_x}\). Then, since \(\|M_{k-1}\|_2 \geq \|M_k\|_2\) and therefore \(\lambda_{k+1}, \ldots, \lambda_d < (1 - \delta_x)\|M_{k-1}\|_2\), we can apply Theorem 4.1.a of the current \(k\) to deduce that \(\|V_k^TU_{>k}\|_2 \leq \varepsilon\) where \(U_{>k} \equiv (u_{k+1}, \ldots, u_d)\). We now apply Lemma B.2 with \(V_s = V_k\) and \(U = U_{>k}\), we obtain a matrix \(Q_k \in \mathbb{R}^{d \times k}\) whose columns are spanned by \(u_1, \ldots, u_k\) and satisfy
\[
\left\| \left( I - Q_kQ_k^T \right) M \left( I - Q_kQ_k^T \right) - \left( I - V_kV_k^T \right) M \left( I - V_kV_k^T \right) \right\|_2 < 16\lambda_1\varepsilon.
\]

However, our assumption says that the second matrix \((I - V_kV_k^T) M (I - V_kV_k^T)\) has spectral norm at least \(\lambda_{k+1}/(1 - \delta_x)\), but we know that \((I - Q_kQ_k^T) M (I - Q_kQ_k^T)\) has spectral norm exactly \(\lambda_{k+1}\) due to the definition of \(Q_k\). Therefore, we must have \(\frac{\lambda_{k+1}}{1 - \delta_x} - \lambda_{k+1} \leq 16\lambda_1\varepsilon\) due to triangle inequality.

In other words, by selecting \(\varepsilon\) in Theorem 4.1.a to satisfy \(\varepsilon \leq \frac{\delta_x}{16\lambda_1/\lambda_{k+1}}\) (which is satisfied by our assumption on \(\varepsilon_{\text{pca}}\)), we get a contradiction so can conclude that \(\|M_k\|_2 \leq \frac{\lambda_{k+1}}{1 - \delta_x}\).

(c) We compute that
\[
v_k^TMv_k = v_k^TM_{k-1}v_k \overset{\text{def}}{=} v_k^TM_{k-1}v_k \overset{\text{def}}{=} \frac{v_k^TM_{k-1}v_k}{\|v_k^T(I - V_{k-1}V_{k-1}^T)v_k\|_2} \overset{\text{def}}{=} \frac{v_k^TM_{k-1}v_k}{1 - \varepsilon_{\text{pca}}} \overset{\text{def}}{=} \frac{(1 - \delta_x/2)\|M_{k-1}\|_2}{(1 - \delta_x)\|M_{k-1}\|_2}.
\]
Above, $\mathbb{1}$ is because $v_k'$ is the projection of $v_k'$ into $V_{k-1}^\perp$, $\mathbb{2}$ is because $\|V_{k-1}^\top v_k'\|^2 \leq \varepsilon_{\text{pca}}$ following the same reason as (C.1) and $\mathbb{3}$ is owing to Theorem 3.1. Next, since $\|M_{k-1}\|_2 \geq \lambda_k$, we automatically have $v_k' M v_k \geq (1 - \delta_x)\lambda_k$. On the other hand, $v_k^\top M v_k = v_k^\top M_{k-1} v_k \leq \|M_{k-1}\|_2 \leq \frac{\lambda_k}{1-\delta_x}$ where the last inequality is owing to Theorem 4.1.b.

(d) Since $\|V_k^\top U\|_2 \leq \varepsilon_c \overset{\text{def}}{=} 8(\lambda_1/\lambda_k)^{1/2}k^{1/4}\varepsilon_{\text{pca}}^{1/2}$ from the analysis of Theorem 4.1.a, we can apply Lemma B.2 to obtain a (column) orthogonal matrix $Q_k \in \mathbb{R}^{d \times k}$ such that

$$M_k' - M_k\|_2 \leq 16\lambda_1 \varepsilon_c,$$

where $M_k' \overset{\text{def}}{=} (I - Q_k Q_k^\top) M (I - Q_k Q_k^\top)$ (C.2)

Suppose $U = (u_{d-p+1}, \ldots, u_d)$ is of dimension $d \times p$, that is, there are exactly $p$ eigenvalues of $M$ that are $\leq (1 - \delta_x)\|M_{k-1}\|_2$. Then, the definition of $Q_k$ in Lemma B.2 tells us $U^\top Q_k = 0$ so $M_k'$ agrees with $M$ on all the eigenvalues and eigenvectors $\{((\lambda_j, u_j))\}_{j=d-p+1}^d$ because an index $j$ satisfies $\lambda_j \leq (1 - \delta_x)\|M_{k-1}\|_2$ if and only if $j \in \{d - p + 1, d - p + 2, \ldots, d\}$.

Denote by $\mu_1, \ldots, \mu_{d-k}$ the eigenvalues of $M_k'$ excluding the $k$ zero eigenvalues in subspace $Q_k$, and assume without loss of generality that $\{\mu_1, \ldots, \mu_p\} = \{\lambda_{d-p+1}, \ldots, \lambda_d\}$. Then,

$$\|M_k'\|_S^q = \sum_{i=1}^{d-k} \mu_i^q = \sum_{i=1}^d \mu_i^q + \sum_{i=d-p+1}^{d-k} \mu_i^q = \sum_{i=d-p+1}^d \lambda_i^q + \sum_{i=d-p+1}^{d-k} \mu_i^q \leq \sum_{i=d-p+1}^d \lambda_i^q + (d-k)\|M_k'\|_2^q$$

$$\leq \sum_{i=d-p+1}^d \lambda_i^q + (d-k)\|M_k\|_2^q + 16\lambda_1 \varepsilon_c$$

Above, $\mathbb{1}$ is because each $\mu_i$ is no greater than $\|M_k'\|_2$, and $\mathbb{2}$ is owing to (C.2) and $\mathbb{3}$ is because of Theorem 4.1.b. Suppose we choose $\varepsilon_c$ so that $\varepsilon_c \leq \frac{\lambda_k + \delta_x}{16\lambda_1}$ (and this is indeed satisfied by our assumption on $\varepsilon_{\text{pca}}$), then we can continue and write

$$\|M_k'\|_S^q \leq \sum_{i=d-p+1}^d \lambda_i^q + (d-k)\|M_k\|_2^q \sum_{i=k+1}^d \lambda_i^q$$

$$\leq \sum_{i=d-p+1}^d \lambda_i^q + \frac{(1 + \delta_x)^q}{(1 - \delta_x)^2} \lambda_k + \sum_{i=k+1}^d \lambda_i^q \leq \frac{(1 + \delta_x)^q}{(1 - \delta_x)^2} \lambda_k + \sum_{i=k+1}^d \lambda_i^q$$

Above, $\mathbb{4}$ is because for each eigenvalue $\lambda_i$ where $i \in \{k + 1, k + 2, \ldots, d - p\}$, we have $\lambda_i > (1 - \delta_x)\|M_{k-1}\|_2 \geq (1 - \delta_x)\lambda_k \geq (1 - \delta_x)\lambda_{k+1}$. Finally, using (C.2) again we have

$$\|M_k\|_S \leq \|M_k'\|_S + \|M_k - M_k'\|_S \leq \|M_k'\|_S + d^{1/p}\|M_k - M_k'\|_2 \leq \frac{1 + \delta_x}{(1 - \delta_x)^2} \left( \sum_{i=k+1}^d \lambda_i^q \right)^{1/q} + 16d^{1/p}\lambda_1 \varepsilon_c$$

As long as $\varepsilon_c \leq \frac{\delta_x \lambda_{k+1}}{16d^{1/p}\lambda_1}$, we have

$$\|M_k\|_S \leq \frac{(1 + \delta_x)^2}{(1 - \delta_x)^2} \left( \sum_{i=k+1}^d \lambda_i^q \right)^{1/q}$$

as desired. Finally, we note that $\varepsilon_c \leq \frac{\delta_x \lambda_{k+1}}{16d^{1/p}\lambda_1}$ is satisfied with our assumption on $\varepsilon_{\text{pca}}$. □
C.1 Proofs of Corollary 4.3 and Corollary 4.4

We first note that since LazySVD outputs $v_i$ one by one, although we have only stated Theorem 4.1 for the last iteration $k$, the claimed properties (a)-(d) hold for all intermediate iterations $s = 1, \ldots, k$.

**Proof of Corollary 4.3 from Theorem 4.1.** By Theorem 4.1.a, we have: $\|V_k^T U\|_2 \leq \epsilon$ where $U = (u_j, \ldots, u_d)$ is a (column) orthonormal matrix and $j$ is the smallest index satisfying $\lambda_j \leq (1 - \delta_x)\|M_{k-1}\|_2$. Since it satisfies $\|M_{k-1}\|_2 \geq \lambda_k$, we have

$$\lambda_{k+1} = \sigma_{k+1}^2 = \sigma_k^2 (1 - \text{gap})^2 \leq \lambda_k (1 - \delta_x) \leq (1 - \delta_x)\|M_{k-1}\|_2,$$

where the first inequality is because our choice of $\delta_x = \text{gap}$. Therefore, $j$ must be equal to $k + 1$ according to its definition, and we conclude $\|V_k^T W\|_2 \leq \epsilon$.

The running time of the algorithm comes directly from Theorem 4.2 by putting in the parameters.

**Proof of Corollary 4.4 from Theorem 4.1.** Denote

$$M_k = (I - V_k V_k^T) M (I - V_k V_k^T) = (I - V_k V_k^T) A A^T (I - V_k V_k^T).$$

According to Theorem 4.1, we have:

$$\|M_k\|_{\infty} \leq \left( \frac{1 + \delta_x}{1 - \delta_x} \right)^2 \|M - M_k^*\|_{\infty} \quad \forall q \geq 1 \text{ and } q = \infty,$$

where $M_k^* = (A_k^*)^T (A_k^*)$ is the rank-$k$ SVD of $M$, and recall $\delta_x = \frac{5}{12}$. For the spectral norm guarantee, we take $q = \infty$ and compute

$$\|A - A_k\|_2 = \|(I - V_k V_k^T) A\|_2 = \sqrt{\|(I - V_k V_k^T) A A^T (I - V_k V_k^T)\|_2}$$

$$= \sqrt{\|M_k\|_2} = \sqrt{\|M_k\|_{\infty}} \leq \frac{1 + \delta_x}{1 - \delta_x} \sqrt{\|M - M_k^*\|_{\infty}}$$

$$\leq (1 + \epsilon) \sqrt{\|M - M_k^*\|_{\infty}} = (1 + \epsilon) \sqrt{\|M - M_k^*\|_2} = (1 + \epsilon) \sigma_{k+1} = (1 + \epsilon)\|A - A_k^*\|_2.$$

For the Frobenius norm guarantee, we take $q = 1$ and compute

$$\|A - A_k\|_F = \sqrt{\text{Tr}[(I - V_k V_k^T) A A^T (I - V_k V_k^T)]} = \sqrt{\|M_k\|_{S_1}} \leq \frac{1 + \delta_x}{1 - \delta_x} \sqrt{\|M - M_k^*\|_{S_1}}$$

$$\leq (1 + \epsilon) \sqrt{\|M - M_k^*\|_{S_1}} = (1 + \epsilon) \sqrt{\sum_{i=k+1}^d \sigma_i^2} = (1 + \epsilon)\|A - A_k^*\|_F.$$

The Rayleigh quotient guarantees directly follow from Theorem 4.1.c. The running time of the algorithm comes directly from Theorem 4.2 by putting in the parameters.

D Proof Details for Our NNZ Running-Time Results

We state and prove a simple proposition first, and then divide this sections into three subsections: Section D.1 deals with column sampling together with the spectral-norm guarantee; Section D.2 deals with column sampling together with the Frobenius-norm guarantee; and Section D.3 deals with entry-wise sampling together with the spectral-norm guarantee.
Proposition D.1. Let \( A, A' \in \mathbb{R}^{d \times n} \) be two matrices with \( d \leq n \), and \( \eta \geq 0 \) be an non-negative real. Suppose \( \| A - A' \|_2 \leq \eta \), then for every \( k \in [d] \), \( \sigma_k(A') - \eta \leq \sigma_k(A) \leq \sigma_k(A') + \eta \)

Proof of Proposition D.1. By symmetry it is enough to show that \( \sigma_k(A') \leq \sigma_k(A) + \eta \).

Let \( v_1, \ldots, v_d \) be the (left) singular vectors of \( A \) in decreasing order of the corresponding singular values, and let \( S_k \) be the space spanned by \( v_k, \ldots, v_d \). Then, for every \( x \in S_k \) that has \( \| x \|_2 = 1 \),

\[
\| x^T A' \|_2 \leq \| x^T (A - A') \|_2 + \| x^T A \|_2 \leq \sigma_k(A) + \eta.
\]

Recall that the Courant-Fischer theorem says that

\[
\sigma_k(A') = \min_{S, \text{dim}(S) = d - k + 1} \max_{x \in S, \| x \|_2 = 1} \| x^T A' \|_2.
\]

Take \( S = S_k \), we immediately obtain \( \sigma_k(A') \leq \sigma_k(A) + \eta \). \(\square\)

D.1 Column Sampling with Spectral-Norm Guarantee

We first state a concentration bound on column sampling (which is easily provable using for instance [22, Theorem 6.6.1]):

Lemma D.2 (column sampling). Let \( A \in \mathbb{R}^{d \times n} \) be a matrix and \( A_i \in \mathbb{R}^d \) be the \( i \)-th column of \( A \). Setting \( p_i \overset{\text{def}}{=} \| A_i \|_2^2 / \| A \|_F^2 \) for each \( i \), and define random rank-1 matrix \( R = \frac{1}{p_i} A_i A_i^\top \) with probability \( p_i \) for each \( i \). For every \( m \geq 1 \), define \( \overline{R}_m \) to be the average of \( m \) independent copies of \( R \), that is, \( \overline{R}_m \overset{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} R_i \) where each \( R_i \) is drawn from \( R \). Then, for every \( \eta, \delta > 0 \), if

\[
m \geq \frac{8\| A \|_2^2 \| A \|_F^2 \log \frac{1}{\delta}}{\eta^2} + \frac{8\| A \|_F^2 \log \frac{1}{3}}{\eta},
\]

we have that with probability \( 1 - \delta \), it satisfies \( \| \overline{R}_m - A A^\top \|_2 \leq \eta \).

The next lemma translates the approximate solution on the column sampled matrix into a spectral guarantee on the original matrix.

Lemma D.3. Let \( A \in \mathbb{R}^{d \times n} \) be a matrix and define \( \overline{R}_m \) as in Lemma D.2. For every \( k \in [d - 1] \), every \( \varepsilon > 0 \), every \( p \in (0, 1) \), and every \( \delta_x \in (0, 1) \), if \( m \geq \frac{32k \log(1/p)\sigma_k(A)^4}{\varepsilon^2 \delta_{k+1}(A)^4} \) and one obtains an \( \delta_x \)-approximate k-SVD of \( \overline{R}_m \) in terms of spectral norm, that is

a column orthogonal matrix \( V \in \mathbb{R}^{d \times k} \) such that \( \| (I - V V^\top) \overline{R}_m (I - V V^\top) \|_2 \leq \frac{\lambda_{k+1}(\overline{R}_m)}{1 - \delta_x} \).

Then, with probability at least \( 1 - p \), this matrix \( V \) satisfies

\[
\| A - V V^\top A \|_2 \leq \frac{\| A - A_k^* \|_2 + 2\varepsilon \| A - A_k^* \|_F}{1 - \delta_x}.
\]
Proof of Lemma D.3. If we let $\eta \overset{\text{def}}{=} \varepsilon^2 \|A - A_k^*\|_F^2 + \varepsilon \|A - A_k^*\|_F \|A - A_k^*\|_2$, we can compute

\[
\begin{align*}
\frac{8\|A\|_F^2 \|A\|_2 \log \frac{1}{p}}{\eta^2} + \frac{8\|A\|_2^2 \log \frac{1}{p}}{\eta} & \leq \frac{8\|A - A_k^*\|_F^2 \|A - A_k^*\|_2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_F^2} + \frac{8\|A\|_F^2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_F^2} \\
& \leq \frac{8\left(\|A - A_k^*\|_F^2 + \varepsilon k\|A\|_2^2\right) \|A\|_2^2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_F^2} + \frac{8\left(\|A - A_k^*\|_F^2 + \varepsilon k\|A\|_2^2\right) \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_F^2} \\
& \leq \frac{8\|A\|_2^2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_2^2} + \frac{8\|A\|_2^2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_2^2}.
\end{align*}
\]

Therefore, according to Lemma D.2, with probability at least $1 - p$, it satisfies

\[
\|AA^\top - \overline{R}_m\|_2 \leq \eta = \varepsilon^2 \|A - A_k^*\|_F^2 + \varepsilon \|A - A_k^*\|_F \|A - A_k^*\|_2.
\]

This further implies that, owing to $\|I - VV^\top\|_2 \leq 1$,

\[
\|(I - VV^\top)AA^\top (I - VV^\top)\|_2 \leq \|(I - VV^\top)\overline{R}_m (I - VV^\top)\|_2 + \|AA^\top - \overline{R}_m\|_2 \\
\leq \frac{\lambda_{k+1}(\overline{R}_m)}{1 - \delta_x} + \eta \leq \frac{\lambda_{k+1}(AA^\top)}{1 - \delta_x} + 2\eta \\
= \frac{\sigma_{k+1}(A)^2 + 2\varepsilon^2 \|A - A_k^*\|_F^2 + 2\varepsilon \|A - A_k^*\|_F \|A - A_k^*\|_2}{1 - \delta_x} \\
\leq \frac{\left(\|A - A_k^*\|_2 + 2\varepsilon \|A - A_k^*\|_F\right)^2}{1 - \delta_x}.
\]

Therefore,

\[
\|(I - VV^\top)A\|_2 = \sqrt{\|(I - VV^\top)AA^\top (I - VV^\top)\|_2} \leq \frac{\|A - A_k^*\|_2 + 2\varepsilon \|A - A_k^*\|_F}{1 - \delta_x}.
\]

Using the previous lemma, it is not hard to deduce our main result of this sub-section.

Theorem D.4. Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$. For every $\varepsilon \in (0, 1/2)$, let $\overline{R}_m$ be the subsampled version of $A$ as defined in Lemma D.2 with $m = \Omega\left(\frac{k\log(1/\varepsilon)\sigma_1^4}{\varepsilon^2 \sigma_{k+1}^4}\right)$.

Then, one can call \textbf{LazySVD} with appropriately chosen $\delta_x$ to produce a matrix $V_k \in \mathbb{R}^{d \times k}$ satisfying

\[
\|A - V_k V_k^\top A\|_2 \leq \|A - A_k^*\|_2 + O(\varepsilon)\|A - A_k^*\|_F ,
\]

and the total running time is $O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^2 d(\sigma_1/\sigma_{k+1})^4}{\varepsilon^2 \sigma_{k+1}^4}\right)$ if AGD is used as the approximate matrix inversion algorithm $A$. Furthermore, if $\varepsilon \leq \frac{1}{2}\|A - A_k^*\|_F / \|A - A_k^*\|_2$, then one can use accelerated SVRG as $A$ and improve the running time to $O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^2 d(\sigma_1/\sigma_{k+1})^4}{\varepsilon^2}\right)$.

Proof of Theorem D.4. Let us define $\overline{M} \overset{\text{def}}{=} \overline{R}_m / \|A\|_F^2$, and we can write $\overline{M} = \frac{1}{m} \sum_{i=1}^m a_i a_i^\top$ where each $a_i$ has Euclidean norm at most 1 due to our definition of $\overline{R}_m$ in Lemma D.2. We pass this matrix $\overline{M}$ as the input $M$ to \textbf{LazySVD}, and it satisfies $\|\overline{M}\|_2 \leq \text{Tr}(\overline{M}) \leq 1$. 

24
Before specifying the parameter choices for $\delta_\times$ and the algorithm choice for $A$, we notice that for sufficiently small $\varepsilon$, Theorem 4.1.b implies $\| (I - V_k V_k^\top) R_m (I - V_k V_k^\top) \|_2 \leq \frac{1}{1 - \delta_\times} \lambda_{k+1}(R_m)$ where $V_k$ is the output matrix from LazySVD. Applying Lemma D.3, we have

$$\| A - V_k V_k^\top A \|_2 \leq \frac{\| A - A_k^\star \|_2 + 2\varepsilon \| A - A_k^\star \|_F}{1 - \delta_\times}.$$ 

Now there are two cases. If we use AGD as the method $A$, then we can choose $\delta_\times = \varepsilon$. In such a case it is easy to see that $\frac{\| A - A_k^\star \|_2 + 2\varepsilon \| A - A_k^\star \|_F}{1 - \delta_\times} \leq \| A - A_k^\star \|_2 + 4\varepsilon \| A - A_k^\star \|_F$ so the guarantee (D.1) is satisfied. The total running time is $O(\text{nnz}(A))$ to sample $\{a_1, \ldots, a_m\}$ plus $\tilde{O}(kmd/\sqrt{\varepsilon})$ to perform LazySVD (see Theorem 4.2).

If we use accelerated SVRG as the method $A$, then we obtain a better dependence on $\varepsilon$ as follows. Suppose $\varepsilon \leq \frac{\| A - A_k^\star \|_2}{2\| A - A_k^\star \|_F}$, and we choose $\delta_\times \leq \frac{\| A - A_k^\star \|_2}{\| A - A_k^\star \|_F} \leq 1/2$. In such a case, one can verify again that $\frac{\| A - A_k^\star \|_2 + 2\varepsilon \| A - A_k^\star \|_F}{1 - \delta_\times} \leq \| A - A_k^\star \|_2 + O(\varepsilon) \| A - A_k^\star \|_F$ so the guarantee (D.1) is satisfied. As for the running time, in addition to $O(\text{nnz}(A))$ for sampling, we need (using Theorem 4.2 again)

$$\tilde{O}(kmd + \frac{km^{3/4} d}{\lambda_k(M)^{1/4} \delta_\times^{1/2}}) = \tilde{O}(kmd + \frac{km^{3/4} d}{\lambda_k(A)^{1/4} \delta_\times^{1/2}}) = \tilde{O}(kmd + \frac{km^{3/4} d}{\delta_\times^{1/2}} \left( \frac{\| A \|_F^2}{\sigma_k(A)^2} \right)^{1/4})$$

$$= \tilde{O} \left( kmd + \frac{km^{3/4} d}{\varepsilon^{1/2}} \left( \frac{\sigma_{k+1}(A)^2 \| A \|_F^2}{\sigma_k(A)^2 \| A - A_k^\star \|_F} \right)^{1/4} \right)$$

Since it can be verified (similar to the proof of Lemma D.3) $\frac{\sigma_{k+1}(A)^2 \| A \|_F^2}{\sigma_k(A)^2 \| A - A_k^\star \|_F} \leq 1 + \frac{\sigma_1(A)^2}{\sigma_k(A)^2} \cdot k$, we conclude that the above running time becomes

$$\tilde{O} \left( kmd + \frac{km^{3/4} d}{\varepsilon^{1/2}} \left( \frac{\sigma_1(A)^2}{\sigma_k(A)^2} \cdot k \right)^{1/4} \right) = \tilde{O} \left( \frac{kmd(\sigma_1/\sigma_{k+1})^4}{\varepsilon^2} \right)$$

where the last equality follows from the definition of $m$. \hfill \Box

### D.2 Column Sampling with Frobenius-Norm Guarantee

The next lemma translates the approximate solution on the column sampled matrix into a Frobenius-norm guarantee on the original matrix.

**Lemma D.5.** Let $A \in \mathbb{R}^{d \times n}$ be a matrix and define $R_m$ as in Lemma D.2. For every $k \in [d - 1]$, every $\varepsilon > 0$, every $p \in (0, 1)$, and every $\delta_\times \in (0, 1)$, if $m \geq \frac{128k^3 \log(1/p) \sigma_1(A)^4}{\varepsilon^2 \sigma_{k+1}(A)^4}$ and one obtains a $\delta_\times$-approximate $k$-SVD of $R_m$ in terms of Rayleigh quotient, that is, a column orthonormal matrix $V = (v_1, \ldots, v_k) \in \mathbb{R}^{d \times k}$ such that

$$\forall i \in [k], \quad |v_i^\top R_m v_i - \lambda_i(R_m)| \leq \delta_\times \lambda_i(R_m).$$

Then, with probability at least $1 - p$, this matrix $V$ satisfies

$$\| (I - VV^\top) A \|_F^2 \leq \delta_\times \| A_k^\star \|_F^2 + (1 + \varepsilon) \| A - A_k^\star \|_F^2$$

**Proof of Lemma D.5:** Using similar arguments as in the proof of Lemma D.3, one can deduce that our choice of $m$ ensures

$$\| AA^\top - R_m \|_2 \leq \eta \overset{\text{def}}{=} \frac{\varepsilon}{2k} \| A - A_k^\star \|_F^2.$$
Which implies that $|\lambda_i(R_m) - \sigma_i^2| \leq \eta$ due to Proposition D.1. Finally,

$$
\|(I - VV^\top)A\|^2_F = \text{Tr}[(I - VV^\top)AA^\top(I - VV^\top)] = \text{Tr}[AA^\top - VAA^\top V]
$$

$$
= \sum_{i=1}^d \sigma_i(A)^2 - \sum_{i=1}^k v_i^\top AA^\top v_i \leq \sum_{i=1}^d \sigma_i(A)^2 + k\eta - \sum_{i=1}^k v_i^\top R_m v_i
$$

$$
\leq \sum_{i=1}^d \sigma_i(A)^2 + k\eta - (1 - \delta_X)(\sigma_i^2(A) - \eta) \leq \sum_{i=1}^k \delta_X \sigma_i(A)^2 + 2k\eta + \sum_{i=k+1}^d \sigma_i(A)^2
$$

$$
= \delta_X \|A_k^*\|^2_F + \|A - A_k^*\|^2_F + 2k\eta = \delta_X \|A_k^*\|^2_F + (1 + \varepsilon)\|A - A_k^*\|^2_F.
$$

Using the previous lemma, it is not hard to deduce our main result of this sub-section.

**Theorem D.6.** Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$. For every $\varepsilon \in (0, 1/2)$, let $R_m$ be the subsampled version of $A$ as defined in Lemma D.2 with $m = \Omega\left(\frac{k^3 \log(1/\varepsilon) \sigma_k^2}{\varepsilon^2 \sigma_{k+1}^2}\right)$. Then, one can call LazySVD with appropriately chosen $\delta_X$ to produce a matrix $V_k \in \mathbb{R}^{d \times k}$ satisfying

$$
\|A - V_k V_k^\top A\|_2 \leq (1 + O(\varepsilon))\|A - A_k^*\|_F,
$$

and the total running time is $O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^4 (\sigma_1/\sigma_{k+1})^5}{\varepsilon^2}\right)$ if AGD is used as the approximate matrix inversion algorithm $A$, or $O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^4 (\sigma_1/\sigma_{k+1})^{4.5}}{\varepsilon^2}\right)$ if $A$ is accelerated SVRG.

**Proof.** One can choose $\delta_X = \varepsilon \|A - A_k^*\|^2_F / \|A_k^*\|^2_F \geq \varepsilon \frac{\sigma_{k+1}^2}{\sigma_k^2}$ as the parameter of LazySVD and the proof is completely analogous to that of Theorem D.4. \qed

### D.3 Entry-Wise Sampling with Spectral-Norm Guarantee

We first state a concentration bound on entry-wise sampling:

**Lemma D.7** (entry-wise sampling [10]). Let $A \in \mathbb{R}^{d \times n}$ be a matrix. Define random single-entry matrix $R = \frac{1}{p_{i,j}} A_{i,j}$ where $(i, j)$ is selected from $[d] \times [m]$ each with probability $p_{i,j} = A_{i,j}^2 / \|A\|^2_F$. For every $m \geq 1$, define $R_s$ to be the average of $s$ independent copies of $R$, that is, $R_s = \frac{1}{s} \sum_{t=1}^s R_t$ where each $R_t$ is drawn from $R$. Then, for every $\eta, p \in (0, 1)$, if

$$
s \geq \frac{28(d + n) \log(2/p)}{\eta^2},
$$

we have with probability $1 - p$, it satisfies $\|A - R_s\|_2 \leq \eta$.

The next lemma translates the approximate solution on the entry-wise sampled matrix into a spectral guarantee on the original matrix.

**Lemma D.8.** Let $A \in \mathbb{R}^{d \times n}$ be a matrix and define $R_s$ as in Lemma D.7. For every $k \in [d - 1]$, every $\varepsilon > 0$, every $p \in (0, 1)$, and every $\delta_X \in (0, 1)$, if $s \geq \frac{56k(d + n) \log(2/p) \sigma_1(A)^2}{\varepsilon^2 \sigma_k(A)^2}$ and one obtains a matrix $R'$ satisfying

$$
\|R' - R_s\|_2 \leq \frac{\sigma_{k+1}(R_s)}{1 - \delta_X}.
$$

Then, with probability at least $1 - p$, we also have

$$
\|A - R'\|_2 \leq \frac{\|A - A_k^*\|_2 + 2\varepsilon\|A - A_k^*\|_F}{1 - \delta_X}.
$$
Proof of Lemma D.8. We first compute that
\[ s \geq \frac{56k(d+n) \ln(2/p)\sigma_1(A)^2}{\varepsilon^2 \sigma_{k+1}(A)^2} \]
\[ \geq \frac{28k(d+n) \ln(2/p)\sigma_1(A)^2}{\varepsilon^2 \sigma_{k+1}(A)^2} + \frac{28d \ln(2/p)}{\varepsilon^2} \]
\[ \geq \frac{28(d+n) \ln(2/p)\|A_k^*\|_F^2}{\varepsilon^2 \|A - A_k^*\|_F^2} + \frac{28(d+n) \ln(2/p)\|A - A_k^*\|_F^2}{\varepsilon^2 \|A - A_k^*\|_F^2} \]
\[ = \frac{28(d+n) \ln(2/p)\|A_k^*\|_F^2 + \|A - A_k^*\|_F^2}{\varepsilon^2 \|A - A_k^*\|_F^2} \]
\[ = \frac{28(d+n) \ln(2/p)\|A\|_F^2}{\varepsilon^2 \|A - A_k^*\|_F^2} . \]

Owing to Lemma D.7, with probability 1 − p, it satisfies \( \|A - \widetilde{R}_s\|_2 \leq \eta \) if \( A - A_k^* \). Using Proposition D.1, we know that \( \sigma_{k+1}(\widetilde{R}_s) \leq \sigma_{k+1}(A) + \eta \), which implies that
\[ \|R' - \widetilde{R}_s\|_2 \leq \frac{\sigma_{k+1}(A) + \eta}{1 - \delta} . \]

Finally,
\[ \|R' - A\|_2 \leq \|R' - \widetilde{R}_s\|_2 + \|A - \widetilde{R}_s\|_2 \leq \frac{\sigma_{k+1}(A) + \eta}{1 - \delta} + \eta \leq \frac{\sigma_{k+1}(A) + 2\eta}{1 - \delta} . \]

Using the previous lemma, it is not hard to deduce our main result of this sub-section.

Theorem D.9. Let \( A \in \mathbb{R}^{d \times n} \) be a matrix with singular values \( \sigma_1 \geq \cdots \geq \sigma_d \geq 0 \). For every \( \varepsilon \in (0, 1/2) \), let \( \widetilde{R}_s \) be the entry-sampled version of \( A \) as defined in Lemma D.7 with \( s = \Omega\left(\frac{k(d+n) \log(1/\varepsilon)\sigma_1^2}{\varepsilon^2 \sigma_{k+1}^2}\right) \). Then, one can call \text{LazySVD} with appropriately chosen \( \delta \) to produce a matrix \( V_k \in \mathbb{R}^{d \times k} \) satisfying
\[ \|A - V_k V_k^T A\|_2 \leq \|A - A_k^*\|_2 + O(\varepsilon)\|A - A_k^*\|_F \],
and the total running time is \( O(\text{nnz}(A)) + \widetilde{O}\left(\frac{k^2(n+d)(\sigma_1/\sigma_{k+1})^2}{\varepsilon^2}\right) \) if \( \text{AGD} \) is used as the approximate matrix inversion algorithm \( A \).

Proof. One can choose \( \delta = \varepsilon \) and a completely analogous proof as that of Theorem D.4 gives us the desired spectral guarantee. The running time follows from Theorem 4.2 because \( \text{nnz}(\widetilde{R}_s) = s \). □

References


Figure 2: Performance on dataset email. Relative error (y-axis) vs. running time (x-axis).
(a) amazon, $k = 10$, \textit{Fnorm} metric

(b) amazon, $k = 10$, \textit{rayleigh} metric

(c) amazon, $k = 10$, \textit{spectral} metric

(d) amazon, $k = 10$, \textit{rayleigh(last)} metric

(e) amazon, $k = 20$, \textit{Fnorm} metric

(f) amazon, $k = 20$, \textit{rayleigh} metric

(g) amazon, $k = 20$, \textit{spectral} metric

(h) amazon, $k = 20$, \textit{rayleigh(last)} metric

(i) amazon, $k = 30$, \textit{Fnorm} metric

(j) amazon, $k = 30$, \textit{rayleigh} metric

(k) amazon, $k = 30$, \textit{spectral} metric

(l) amazon, $k = 30$, \textit{rayleigh(last)} metric

Figure 3: Performance on dataset \textit{amazon}. Relative error ($y$-axis) vs. running time ($x$-axis).
Figure 4: Performance on dataset news20. Relative error (y-axis) vs. running time (x-axis).
Figure 5: Performance on dataset rcv1. Relative error (y-axis) vs. running time (x-axis).