COLUMN SUBSET SELECTION WITH OUTLIERS: APPROXIMATION ALGORITHMS USING FEWER COLUMNS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the column subset selection problem with outliers, where the goal is to select a small subset of columns that approximates the input matrix after discarding at most mm outlier columns. The existing bicriteria algorithm utilizes iterative uniform sampling and achieves a strong approximation guarantee, but its column complexity depends on the number of columns d and can be substantially larger than k. A natural approach to reducing the column complexity is to use importance sampling. However, large importance scores need not correspond to columns useful for reconstructing the inliers, since arbitrary outliers may introduce high-leverage directions and dominate the sampling distribution. Directly applying importance sampling may make it difficult to preserve the inlier structure with a small number of selected columns. To address this difficulty, we propose a ridge-leverage peeling method that avoids separating inlier and outlier scores, allowing us to iteratively remove columns without identifying the outliers while controlling the number of discarded inliers. The resulting algorithm achieves a $(1+\varepsilon)$-approximation using $O(k/\varepsilon^2)$ columns while discarding at most $(1+\varepsilon)m$ columns, improving the previous $O\left(\frac{k}{\varepsilon^6}\left(\log\frac{d}{m}+\frac{1}{\varepsilon}\right)\right)$ column bound, where $\varepsilon \in (0, 1/2)$. We further give an fixed-parameter tractable (FPT) algorithm that selects exactly k columns while discarding at most $(1+\varepsilon)m$ columns and achieves an $O(k/\varepsilon)$-approximation.

1 INTRODUCTION

Column subset selection (CSS) is a widely studied matrix approximation problem with applications in dimensionality reduction and feature selection Deshpande et al. (2006); Boutsidis et al. (2009a;b; 2014); Woodruff & Yasuda (2023). Unlike principal component analysis (PCA), which forms new directions from linear combinations of the input columns, CSS selects actual input columns, which can improve interpretability and facilitate downstream analysis Mahoney & Drineas (2009); Boutsidis et al. (2008; 2009b); Drineas et al. (2008). Specifically, given a matrix and an integer k, the goal of CSS is to select k columns whose span provides a good approximation to the input matrix. However, even a small number of arbitrary column outliers can substantially distort the geometry of a high-dimensional matrix and affect column selection Xu et al. (2010). In such settings, standard CSS may use its selected columns to represent outlier directions, while the resulting subspace may fail to preserve the underlying low-dimensional structure of the inliers. This motivates the study of CSS with outliers, where at most m columns can be discarded Bhaskara & Kumar (2018). We formally define the problem as follows.

Column Subset Selection with Outliers. Let $A \in \mathbb{R}^{n\times d}$, and let $k \in [d]$ and $m \geq 1$. The goal of CSS with outliers is to discard at most m columns and select exactly k columns from the remaining matrix such that the selected columns well approximate all remaining columns. We define

$$\min_{\substack{N,S\subseteq[d]\\ |N|\leq m,\ |S|=k,\ N\cap S=\emptyset}} \left\|A_{[d]\setminus N} - \Pi_{A_S} A_{[d]\setminus N}\right\|_F^2,$$

where Π_{A_S} denotes the orthogonal projector onto $\mathrm{span}(A_S)$. For comparison, let $\mathrm{OPT}_{k,m}(A) := \min_{N\subseteq[d],|N|\leq m} \|A_{[d]\setminus N} - (A_{[d]\setminus N})_k\|_F^2$ denote the minimum squared Frobenius error of a rank-k approximation after discarding at most m columns.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The most closely related result is the bicriteria approximation of Bhaskara & Kumar (2018). They propose an iteratively uniform sampling method that uses a guess of the optimal reconstruction error. A sufficiently accurate guess is obtained by considering polynomially many candidate values. In each round, the algorithm uniformly samples columns, removes those well approximated by their span, and repeats on the remaining columns. It achieves a $(1+\varepsilon)$-approximation while discarding at most $(1+\varepsilon)m$ columns, using $O\left(\frac{k}{\varepsilon^6}\left(\log\frac{d}{m}+\frac{1}{\varepsilon}\right)\right)$ selected columns. However, when $m \ll d$, the number of selected columns still grows with the total number of columns d and can be substantially larger than k. A larger selected set also weakens the role of CSS as a compact representation in terms of original features. In applications where selected columns correspond to features that must be acquired or evaluated downstream, retaining more columns can increase measurement and computational costs Guyon & Elisseeff (2003). This issue is particularly relevant in biomarker selection, where the goal is often to identify a small and stable set of genes or other markers for subsequent analysis or validation Abeel et al. (2010). This motivates the goal of retaining a $(1+\varepsilon)$ approximation while reducing the selected-column complexity to depend only on k and ε.

A natural approach toward a selected-column bound independent of d is leverage-based importance sampling, which has been widely used in classical CSS to obtain strong approximation guarantees Cohen et al. (2015; 2017); Woodruff & Yasuda (2023). In the classical setting, the importance scores and the approximation target are both determined by the same input matrix. However, in the presence of outliers, the scores must be computed from the full matrix A, while the desired approximation concerns the unknown inlier matrix $A_{I^\star}$. Therefore, an outlier can receive a large score by introducing a direction that is important for A but irrelevant to approximating $A_{I^\star}$. Leverage-based sampling may consequently assign substantial probability to such outlier columns. At the same time, a large score cannot simply be used to identify an outlier, since an inlier may also carry an important direction of the inlier matrix. Thus, the challenge is to exploit leverage information without identifying outliers, while removing the d-dependence and retaining strong approximation guarantees.

1.1 Our Contributions

In this paper, we propose approximation algorithms for CSS with outliers with reduced selected-column complexity. We first develop a ridge-leverage peeling approach whose selected-column bound depends only on k and ε. We use ridge leverage scores to guide the peeling process without identifying individual columns as inliers or outliers. The main difficulty is that deleting columns can increase the ridge leverage scores of the remaining columns. We show that, at a suitable regularization scale, the surviving optimal inliers nevertheless contribute only $O(k)$ total ridge leverage mass throughout the adaptive peeling process. Therefore, whenever the total mass is large, sampling a column in proportion to its current score removes an optimal inlier with only a small probability. Repeating this procedure reduces the total ridge leverage mass while discarding only a limited number of optimal inliers. Once the total mass is sufficiently small, leverage-based column sampling yields a $(1+\varepsilon)$-approximation using $O(k/\varepsilon^2)$ selected columns while discarding at most $(1+\varepsilon)m$ columns.

A direct implementation of the peeling procedure would be expensive, since ridge leverage scores change after each deletion and repeatedly recomputing them would require many passes over the full matrix. To avoid this cost, we combine a hierarchy of random buckets with low-dimensional score maintenance. The hierarchy processes local instances from the bottom up and repeatedly merges their surviving columns, keeping the ridge leverage mass of each intermediate instance under control. Within each local peeling call, we construct a fixed low-dimensional representation that supports valid score estimates after further deletions. Therefore, score updates and score-proportional peeling can be performed without rescanning all columns after every deletion. The hierarchy alone introduces a logarithmic factor in the remaining ridge leverage mass. We remove this factor through one final global peeling step, which reduces the total mass to $O(k/\varepsilon)$. Then, we apply column sampling together with spectral sparsification to obtain $O(k/\varepsilon^2)$ selected columns. Finally, the columns removed during peeling are used only to construct the reduced instance. The final discarded set is determined by residual trimming on the original matrix after the selected span has been constructed.

We further consider the stricter setting in which exactly k columns must be selected, and give a fixed-parameter tractable (FPT) algorithm with an $O(k/\varepsilon)$ approximation. Peeling and projection first reduce the relevant dimension to $O(k/\varepsilon)$. The main difficulty is that preserving the reconstruction error alone does not control the number of discarded columns. In particular, approximation error introduced by a reduced instance may cause additional columns to cross the removal threshold

Table 1: Comparison of algorithms for CSS with outliers. Here, $A \in \mathbb{R}^{n\times d}$, k is the number of selected columns, ε is the approximation accuracy, and $m \geq 1$ is the maximum number of discarded columns. Both Bhaskara & Kumar (2018) and our algorithms use a guess of the optimal reconstruction error, and we follow their guessing scheme. For our algorithms, we assume $f^\star = \mathrm{OPT}_{k,m}(A) > 0$ and $f^\star \leq \widehat{f} \leq 2f^\star$.

Method	Columns	Outliers	Approximation	Running time
Bhaskara & Kumar (2018)	$O\left(\frac{k}{\varepsilon^6}\left(\log\frac{d}{m}+\frac{1}{\varepsilon}\right)\right)$	$(1+\varepsilon)m$	$1+\varepsilon$	Near-linear[1]
Ours	$O\left(\frac{k}{\varepsilon^2}\right)$	$(1+\varepsilon)m$	$1+\varepsilon$	Near-linear[2]
Ours (exact-k)	k	$(1+\varepsilon)m$	$O(k/\varepsilon)$	FPT[3]

[1] The algorithm in Bhaskara & Kumar (2018) can be implemented in $\widetilde{O}\left(\frac{ndk}{\varepsilon^9}+\frac{nk^2}{\varepsilon^{13}}\right)$, where $\widetilde{O}(\cdot)$ hides polylogarithmic factors in d, k, and $1/\epsilon$.

[2] Our bicriteria algorithm runs in $\widetilde{O}\left(\frac{ndk^2}{\varepsilon^2}+d\left(\frac{k}{\varepsilon}\right)^3+\frac{nk^2}{\varepsilon^6}+\frac{k^4}{\varepsilon^{11}}\right)$.

[3] Our exact-k algorithm runs in $O\left(2^{k^4/\varepsilon^4}d^2+m(n^3+n^2d)+kdn^\omega\log n\right)$, where ω is the matrix multiplication exponent.

when the solution is evaluated on the original matrix. To control both quantities simultaneously, we introduce a truncated-residual objective that combines the reconstruction cost of retained columns with a fixed penalty for every column crossing the removal threshold. A weighted coreset preserves this objective uniformly over candidate subspaces and reduces the remaining search to a parameterized instance. Then, we bound the additional removals introduced by the coreset together with the optimal inliers already removed during peeling. Finally, a standard exact-k CSS algorithm applied to the retained matrix selects exactly k columns while discarding at most $(1+\varepsilon)m$ columns.

Our main results are summarized below. Table 1 compares them with prior work.

- We develop a ridge-leverage peeling algorithm for CSS with outliers that removes the dependence on d from the selected-column complexity. The algorithm achieves a $(1+\varepsilon)$-approximation using $O(k/\varepsilon^2)$ columns while discarding at most $(1+\varepsilon)m$ columns.
- We further give a fixed-parameter tractable (FPT) algorithm for the exact-k setting, parameterized by k and $1/\varepsilon$. The algorithm selects exactly k columns, discards at most $(1+\varepsilon)m$ columns, and achieves an $O(k/\varepsilon)$-approximation.

1.2 Other Related Work

A closely related problem is PCA with outliers, which seeks a low-dimensional subspace after discarding a prescribed number of samples. Fomin, Fedor and Golovach, Petr and Panolan, Fahad and Simonov, Kirill study exact algorithms and approximation hardness, while Dahiya et al. (2021) develop approximation and parameterized algorithms under several regimes. Sampling-based and online approaches have also been considered Deshpande & Pratap (2021); Zhu & Shen (2022), and more recent work studies geometric algorithms under additional structural conditions Hashemian et al. (2026). These works focus on robust subspace approximation or recovery, whereas our goal is to construct the approximation using a small number of original columns, including an exact-k column-selection guarantee.

Another line of work studies robust variants of CSS under entrywise ℓ_p losses. Early work develops approximation algorithms for entrywise ℓ_1 low-rank approximation and column subset based approximations Song et al. (2017), followed by sampling guarantees under low-rank noise models Song et al. (2019) and nearly tight bounds for entrywise ℓ_1 CSS Mahankali & Woodruff (2021). Streaming, distributed, and online variants have also been studied Jiang et al. (2021); Woodruff & Yasuda (2023). In this line of work, robustness is obtained through the choice of loss function, whereas our setting retains the squared Frobenius objective and discards only a bounded number of arbitrary columns.

2 PRELIMINARIES

Notation. For a positive integer d, let $[d] = \{1, \dots, d\}$ and write $A = [a_1, \dots, a_d] \in \mathbb{R}^{n \times d}$. For $S \subseteq [d]$, let A_S denote the submatrix indexed by S. For a matrix B, let $B^\dagger$ be its Moore–Penrose pseudoinverse and $\Pi_B = BB^\dagger$ the orthogonal projector onto its column span. We use $\|\cdot\|_2$ and $\|\cdot\|_F$ for the spectral and Frobenius norms, $\sigma_i(B)$ for the ith largest singular value, and B_k for a best rank-k approximation of B. For symmetric matrices X and Y, we write $X \preceq Y$ if $Y - X$ is positive semidefinite. The notation $\widetilde{O}(\cdot)$ hides polylogarithmic factors in the problem parameters.

Optimal solution and scale estimate. Let $f^\star = \mathrm{OPT}_{k,m}(A)$, as defined in the introduction, and fix an optimal outlier set $N^\star \in \arg\min_{N \subseteq [d], |N| \le m} \|A_{[d]\setminus N} - (A_{[d]\setminus N})_k\|_F^2$. We define $I^\star = [d] \setminus N^\star$, and let $U^\star \subseteq \mathbb{R}^n$ be a subspace of dimension at most k attaining the best rank-k approximation error of $A_{I^\star}$. Let $P^\star$ denote the orthogonal projector onto $U^\star$. We call $I^\star$ the optimal inliers; both $I^\star$ and $P^\star$ are used only in the analysis. Following Bhaskara & Kumar (2018), for $f^\star > 0$ we use an estimate $\widehat{f}$ satisfying $f^\star \le \widehat{f} \le 2f^\star$, and set $\lambda = \widehat{f}/k$.

Ridge leverage scores. For $B = [b_1, \dots, b_s] \in \mathbb{R}^{n \times s}$ and $\lambda > 0$, define the λ-ridge leverage score Cohen et al. (2017) by $\tau_j^\lambda(B) = b_j^\top (BB^\top + \lambda I_n)^{-1} b_j$. Its total mass is $Z_\lambda(B) = \sum_{j=1}^s \tau_j^\lambda(B) = \sum_i \frac{\sigma_i(B)^2}{\sigma_i(B)^2 + \lambda}$. We use the standard bounds $0 \le \tau_j^\lambda(B) \le 1$ and $Z_\lambda(B) \le k + \|B - B_k\|_F^2/\lambda$. For $R \subseteq [d]$ and $j \in R$, we have $\tau_j^R = a_j^\top (A_R A_R^\top + \lambda I_n)^{-1} a_j$.

3 BICRITERIA APPROXIMATION WITH FEWER COLUMNS

In this section, we present our bicriteria algorithm, which achieves a $(1+\varepsilon)$-approximation using $O(k/\varepsilon^2)$ selected columns while discarding at most $\lceil (1+\varepsilon)m \rceil$ columns. We focus on $0 < \varepsilon \le 1/2$ and the positive-error case $0 < f^\star \le \widehat{f} \le 2f^\star$.

Algorithm overview. The key idea behind Algorithm 2 is to use peeling to prepare the matrix for column selection, not to identify the final outliers. For any current column set, the surviving optimal inliers contribute only $O(k)$ total ridge leverage mass (Lemma 3.1). Therefore, when the total mass is large, sampling according to the current score estimates removes an optimal inlier with small probability. The routine RIDGEPEEL repeats this process until the prescribed mass threshold is reached. Within each call, a fixed low-dimensional representation supports score estimates that remain valid under subsequent deletions (Lemma 3.2). A separate auxiliary tree supports sampling from the changing distribution without recomputing all individual scores; its construction and update analysis are deferred to the appendix.

To keep the local representations small, the algorithm organizes the peeling calls hierarchically. For $m \ge 1$, the columns are randomly partitioned into buckets and processed through a balanced binary tree. Each local call reduces its surviving set to ridge leverage mass at most L. Therefore, merging two child sets gives an input mass at most $2L$ for the next call, keeping the local representations controlled throughout the hierarchy. After the root is processed, a final peeling call uses the smaller target $K = O(k/\varepsilon)$ to remove the logarithmic factor in L from the final column bound. Theorem 3.3 shows that the resulting set R satisfies $Z_\lambda(A_R) \le K$ while the entire peeling stage removes at most $\varepsilon m/10$ optimal inliers with constant probability.

Starting from A_R, FASTCANDIDATECOVER samples and reweights columns to preserve its regularized covariance, and TOPTAILONCOVER applies spectral sparsification to select $O(k/\varepsilon^2)$ original columns S. Both routines are deferred to the appendix. Lemma 3.4 gives a spectral reconstruction guarantee for A_R, and hence for its unknown surviving optimal inliers. Finally, the algorithm evaluates residuals on the original matrix and lets N contain the largest residuals outside S, up to $M = \lceil (1+\varepsilon)m \rceil$. Thus, the peeling deletions D are used only to construct the reduced instance, while the final discarded set is determined after the selected span has been constructed. Algorithm 2 summarizes the complete procedure.

We set $\lambda = \widehat{f}/k$, $M = (1+\varepsilon)m$, and $K = 7200k/\varepsilon$. Let D denote the columns removed during peeling and $R = [d] \setminus D$. The peeling stage aims to ensure $Z_\lambda(A_R) \le K$ and $|D \cap I^\star| \le \varepsilon m/10$.

Algorithm 1 RIDGEPEEL

Input: column block A_U indexed by $U \subseteq [d]$, ridge parameter $\lambda > 0$, target mass $L > 0$, an upper bound $H \geq \max\{1, Z_\lambda(A_U)\}$, failure probability $\eta \in (0, 1)$, and removed indices D, and the total removal limit M.
Output: surviving indices $R \subseteq U$ and the updated set D, or FAIL.

if $U = \emptyset$, $A_U = 0$, or $H \leq L$ **then**
  return (U, D).
end if
Use ridge sampling to construct $V \in \mathbb{R}^{n \times s}$ with $V^\top V = I_s$, $s = O(H \log((H+1)/\eta))$, and $\|(I_n - VV^\top)A_U\|_2^2 \leq \lambda$, with failure probability at most η.
if preprocessing returns FAIL **then**
  return FAIL.
end if
For each $j \in U$, compute $x_j \leftarrow V^\top a_j$ and $t_j \leftarrow \|(I - VV^\top)a_j\|_2^2/\lambda$.
Set $G \leftarrow \sum_{j \in U} x_j x_j^\top$ and $T \leftarrow \sum_{j \in U} t_j$.
Initialize $R \leftarrow U$ and $P \leftarrow (G + \lambda I_s)^{-1}$.
Set $Q \leftarrow 3(\mathrm{tr}(PG) + T)$.
while $Q > L$ **do**
  if $|D| = M$ **then**
    return FAIL.
  end if
  Sample $J \in R$ with probability $\Pr[J = j] = 3(x_j^\top P x_j + t_j)/Q$ for each $j \in R$.
  Set $y \leftarrow P x_J$ and update $P \leftarrow P + yy^\top/(1 - x_J^\top y)$.
  Update $G \leftarrow G - x_J x_J^\top$ and $T \leftarrow T - t_J$.
  Update $R \leftarrow R \setminus \{J\}$ and $D \leftarrow D \cup \{J\}$.
  Set $Q \leftarrow 3(\mathrm{tr}(PG) + T)$.
end while
return (R, D).

The first bound reduces the ridge leverage mass to $O(k/\varepsilon)$, which enables the subsequent selection of $O(k/\varepsilon^2)$ columns. The second limits the number of optimal inliers lost during peeling, leaving sufficient slack within the final outlier budget M.

3.1 HIERARCHICAL RIDGE-LEVERAGE PEELING

We first show that ridge-leverage peeling can reduce the total mass without removing too many optimal inliers. Although an individual optimal inlier may have a large ridge leverage score, the total mass of all surviving optimal inliers remains bounded throughout the peeling process.

Lemma 3.1. *For every $R \subseteq [d]$, the total ridge leverage mass of the surviving optimal inliers satisfies $\sum_{j \in R \cap I^\star} \tau_j^R \leq 2k$.*

Lemma 3.1 directly gives a deletion guarantee. Suppose the current estimates satisfy $\tau_j^R \leq q_j^R \leq 9\tau_j^R$, and let $Q_R = \sum_{j \in R} q_j^R$. The optimal inliers contribute at most $18k$ estimated mass. Let $\mathcal{F}_{t-1}$ denote the sigma-field generated by the first $t-1$ deletions, and let R_t be the surviving set before the tth deletion. Whenever $Q_{R_t} > L$, we have

$$\Pr[j_t \in I^\star \mid \mathcal{F}_{t-1}] = \frac{\sum_{j \in R_t \cap I^\star} q_j^{R_t}}{Q_{R_t}} \leq \frac{18k}{L} \leq \frac{\varepsilon}{400}. \tag{1}$$

The remaining issue is that ridge leverage scores change after every deletion. Recomputing all scores from the current matrix would make peeling expensive. We construct a fixed low-dimensional representation for each peeling call that provides constant-factor estimates under subsequent deletions.

Maintaining scores after deletions. For each block U, we construct a low-dimensional representation that remains fixed throughout the peeling call. Suppose $Z_\lambda(A_U) \leq H$. Constant-accuracy ridge

Algorithm 2 RobustCSS

Input: matrix $A \in \mathbb{R}^{n\times d}$, target rank $k \in [d]$, outlier budget $m \in \{0,\dots,d\}$, accuracy $\varepsilon \in (0, 1/2]$, and an estimate $\widehat{f}$ satisfying $0 < f^\star \le \widehat{f} \le 2f^\star$.
Output: selected column indices S and discarded column indices N, or FAIL.

Set $\rho \leftarrow 1/40$, $\lambda \leftarrow \widehat{f}/k$, $M \leftarrow (1+\varepsilon)m$, and $K \leftarrow 7200k/\varepsilon$.
Initialize $D \leftarrow \emptyset$ and $R \leftarrow [d]$.
if $m \ge 1$ **then**
  Set $\ell \leftarrow \lceil \log(4d(m+1)/\rho) \rceil$ and $K_0 \leftarrow \lceil 16\ell \rceil$.
  Set $L \leftarrow C_W k\ell/\varepsilon$ and $\eta \leftarrow \rho/(4d+2)$, where $C_W \ge 7200$ is an absolute constant.
  Independently assign each column uniformly to one of $B_0 = \lceil m/K_0 \rceil$ buckets.
  Build a balanced binary tree with these buckets as leaves, and let r denote its root.
  for nodes v in bottom-up order **do**
    Let U_v be the bucket at a leaf, or the union of its two child survivor sets.
    Set $H_v \leftarrow 2k + 4K_0$ at a leaf and $H_v \leftarrow 2L$ otherwise.
    $(R_v, D) \leftarrow \text{RidgePeel}(A_{U_v}, \lambda, L, H_v, \eta, D, M)$. (See Algorithm 1)
  end for
  $(R, D) \leftarrow \text{RidgePeel}(A_{R_r}, \lambda, K, L, \eta, D, M)$. (See Algorithm 1)
end if
Construct a reweighted candidate set preserving the regularized covariance of A_R by $(C, \widetilde{C}) \leftarrow$ FastCandidateCover$(A_R, \lambda, K, \varepsilon, \rho)$. (See Algorithm 5 in Appendix)
Sparsify the candidate set to $O(k/\varepsilon^2)$ original columns while preserving the required spectral approximation by $S \leftarrow$ TopTailOnCover$(\widetilde{C}, C, \lambda, K, \varepsilon)$. (See Algorithm 6 in Appendix)
For each $j \in [d] \setminus S$, compute $r_j \leftarrow \|(I - \Pi_{A_S})a_j\|_2^2$.
Let N contain the $\min\{M, d - |S|\}$ indices in $[d] \setminus S$ with the largest residuals.
return (S, N).

sampling Cohen et al. (2017) gives, with probability at least $1-\eta$, an orthonormal matrix $V \in \mathbb{R}^{n\times s}$ with $s = O(H\log((H+1)/\eta))$ and $\|(I_n - VV^\top)A_U\|_2^2 \le \lambda$.

For each $j \in U$, we store $x_j = V^\top a_j$ and $t_j = \|(I_n - VV^\top)a_j\|_2^2/\lambda$. These quantities remain fixed as columns are removed. For a surviving set $R \subseteq U$, we maintain $P_R = (X_R X_R^\top + \lambda I_s)^{-1}$, where $X_R = [x_j]_{j\in R}$, and estimate τ_j^R by $q_j^R = 3(x_j^\top P_R x_j + t_j)$. Thus, each deletion only requires updating the low-dimensional matrix P_R.

Lemma 3.2. *Once V satisfies the residual bound above, the resulting estimates satisfy $\tau_j^R \le q_j^R \le 9\tau_j^R$ simultaneously for every $R \subseteq U$ and $j \in R$.*

The simultaneous guarantee is essential because the surviving sets are chosen adaptively. If $Q_R = \sum_{j\in R} q_j^R$, then $Z_\lambda(A_R) \le Q_R$, while the optimal inliers contribute at most $18k$ to Q_R. Therefore, when RidgePeel stops at $Q_R \le L$, the returned set satisfies $Z_\lambda(A_R) \le L$. The dynamic sampling data structure used to sample from the changing scores and its update cost are given in the appendix.

Hierarchical peeling. The low-dimensional representation used by RidgePeel depends on the ridge leverage mass of its input. Applying it directly to the full matrix may require a mass bound of $2k+m$, which can be large when many outliers are present. We therefore randomly partition the columns and organize the peeling calls along a balanced binary tree.

Let $\ell = \lceil \log(4d(m+1)/\rho) \rceil$, $K_0 = \lceil 16\ell \rceil$, and partition the columns into $B_0 = \lceil m/K_0 \rceil$ leaf groups. With constant probability, each leaf contains at most $4K_0$ optimal outliers. Together with Lemma 3.1 and $\tau_j^R \le 1$, this implies that every leaf has ridge leverage mass at most $2k + 4K_0$. We apply RidgePeel to each leaf with the common target $L = C_W k\ell/\varepsilon$. Then, the tree is processed bottom-up. If an internal node has child survivor sets R_{v_1} and R_{v_2}, its input is their union. By subadditivity of the ridge leverage mass,

$$Z_\lambda(A_{R_{v_1}\cup R_{v_2}}) \le Z_\lambda(A_{R_{v_1}}) + Z_\lambda(A_{R_{v_2}}) \le 2L.$$

Thus, every internal call starts from a controlled mass bound and peels the merged set back to L. After the root is processed, we apply RidgePeel once more with the smaller target $K = 7200k/\varepsilon$.

This final call removes the logarithmic factor in L from the mass bound used by the subsequent column-selection step.

It remains to control the optimal inliers removed across the entire hierarchy. By Equation (1), each deletion in a call with target L removes an optimal inlier with conditional probability at most $\varepsilon/400$. The same bound holds for the final call since $18k/K \le \varepsilon/400$. Since at most $M \le 2m$ columns are removed when $m \ge 1$, the expected number X of removed optimal inliers is at most $\varepsilon m/200$. Markov's inequality gives $\Pr[X > \varepsilon m/10] \le 1/20$. Moreover, on the event $X \le \varepsilon m/10$, the removal limit M cannot be reached, since removing M distinct columns would require removing at least $M - |N^\star| \ge \lceil \varepsilon m \rceil$ optimal inliers. Hence, all prescribed mass thresholds are reached while removing at most $\varepsilon m/10$ optimal inliers with constant probability. When $m = 0$, no peeling is required.

Combining the random partition, the deletion-stable score estimates, and the global deletion bound gives the peeling guarantee.

Theorem 3.3. *With constant probability, the peeling stage of Algorithm 2 completes without returning* FAIL *and produces $R = [d] \setminus D$ satisfying $Z_\lambda(A_R) \le K$, and $|D \cap I^\star| \le \varepsilon m/10$.*

The hierarchy also keeps every local preprocessing instance within ridge mass $O(k\ell/\varepsilon)$, while each removed column incurs only one deletion update. The detailed failure-probability and running-time analysis is deferred to the appendix.

3.2 Column Selection and Residual Trimming

The surviving matrix may still contain outliers, so the column-selection guarantee must also apply to its unknown optimal-inlier subset. From $Z_\lambda(A_R) \le K$, ridge sampling constructs a candidate cover with $\widetilde{O}(K/\varepsilon^2)$ columns, and spectral sparsification reduces it to $O(K/\varepsilon)$ original columns.

Lemma 3.4. *On the successful peeling and candidate-sampling events, the column-selection step returns $S \subseteq R$ satisfying $|S| = O(K/\varepsilon) = O(k/\varepsilon^2)$, and $\|(I_n - \Pi_{A_S})A_R\|_2^2 \le \varepsilon\lambda/16$.*

Let $J = R \cap I^\star$ and $Q = I_n - \Pi_{A_S}$. The surviving optimal inliers satisfy

$$\|QA_J\|_F^2 \le \|A_J - (A_J)_k\|_F^2 + k\|QA_J\|_2^2 \le f^\star + \frac{k\varepsilon\lambda}{16} \le (1 + \varepsilon/8)f^\star,$$

where the last inequality uses $\lambda = \widehat{f}/k$ and $\widehat{f} \le 2f^\star$.

We now return to the original matrix and trim the largest residuals outside S. For comparison, we consider

$$\widetilde{N} = \big(N^\star \cup (D \cap I^\star)\big) \setminus S.$$

This set is disjoint from S and satisfies $|\widetilde{N}| \le m + \varepsilon m/10 \le M$. Every column retained after removing $\widetilde{N}$ belongs to either J or S, and the columns in S have zero residual. Hence, its reconstruction error is at most $\|QA_J\|_F^2$. Since the algorithm removes the M largest eligible residuals, its trimmed reconstruction error is no larger. This transfers the guarantee from the peeled matrix back to the original input.

Theorem 3.5. *Suppose $f^\star > 0$ and $f^\star \le \widehat{f} \le 2f^\star$. With constant probability, Algorithm 2 returns disjoint sets $S, N \subseteq [d]$ with $|S| = O(k/\varepsilon^2)$ and $|N| \le (1+\varepsilon)m$. The resulting approximation satisfies*

$$\|(I_n - \Pi_{A_S})A_{[d]\setminus N}\|_F^2 \le (1+\varepsilon)f^\star.$$

The algorithms runs in $\widetilde{O}\left(\frac{ndk^2}{\varepsilon^2} + d\left(\frac{k}{\varepsilon}\right)^3 + \frac{nk^2}{\varepsilon^6} + \frac{k^4}{\varepsilon^{11}}\right)$ time.

4 Exact-k Approximation

We next consider the more restrictive exact-k setting. Algorithm 3 proceeds in three stages. First, exact-score ridge-leverage peeling returns a surviving set $R = [d] \setminus D$ with total ridge leverage mass $O(k/\varepsilon)$ while removing only a limited number of optimal inliers. We then project A_R onto the singular directions whose squared singular values exceed λ. The ridge-mass bound limits the

Algorithm 3 EXACTKROBUSTCSS

Input: matrix $A \in \mathbb{R}^{n\times d}$, target rank k, outliers m, ε, and $\widehat{f}$, where $f^\star \le \widehat{f} \le 2f^\star$.
Output: A set $S \subseteq [d]$ with k column indices, and a set of outliers $N \subseteq [d]$ or **FAIL**.

Set $M \leftarrow (1+\varepsilon)m$, $\Delta \leftarrow M - m$, $\lambda \leftarrow \widehat{f}/k$, and $K \leftarrow C_0 k/\varepsilon$.
Initialize $R \leftarrow [d]$ and $D \leftarrow \emptyset$.
while $Z_\lambda(A_R) > K$ **do**
  if $|D| = M$ **then**
    return FAIL.
  end if
  Compute $\{\tau_j^R\}_{j\in R}$.
  Sample one index $J \in R$ according to $\Pr[J=j] = \frac{\tau_j^R}{Z_\lambda(A_R)}$.
  $R \leftarrow R \setminus \{J\}$, and $D \leftarrow D \cup \{J\}$.
end while
Let $V \in \mathbb{R}^{n\times r}$ contain the left singular vectors of A_R whose squared singular values are larger than λ.
Set $z_j \leftarrow V^\top a_j$ for all $j \in R$, and $\theta \leftarrow \frac{64\widehat{f}}{\Delta}$.
Construct a strong ξ-coreset for $\Phi_\theta(W) \coloneqq \sum_{j\in R} \min\left\{\|(I-\Pi_W)z_j\|_2^2, \theta\right\}$ over all subspaces $W \subseteq \mathbb{R}^r$ with $\dim(W) \le k$, where $\xi \coloneqq \frac{\varepsilon}{256}$.
Solve the coreset problem exactly and obtain $\widehat{W}$.
Set $B \leftarrow \left\{j \in R : \|(I-\Pi_{\widehat{W}})z_j\|_2^2 \ge \theta\right\}$, and $T \leftarrow R \setminus B$.
if $|T| \ge k$ **then**
  Apply VOLUMECSS(A_T, k) from Lemma B.7, obtaining $S \subseteq T$ with $|S| = k$.
else
  Choose any $P \subseteq [d] \setminus T$ of size $k - |T|$, and set $S \leftarrow T \cup P$.
end if
Let $N_M(S)$ contain the $\min\{M, d-k\}$ largest-residual indices in $[d] \setminus S$ with respect to $\mathrm{span}(A_S)$.
return $(S, N_M(S))$.

projected dimension to $O(k/\varepsilon)$ and controls the projection error, reducing the subsequent robust subspace search to a low-dimensional problem.

The projected instance may still contain outliers. The second stage therefore optimizes a truncated-residual objective that caps large squared residuals at a common threshold. A strong coreset reduces this search to a parameterized instance on which exact optimization is fixed-parameter tractable. We enumerate realizable residual-threshold patterns on the coreset, compute a weighted PCA candidate from the below-threshold columns of each pattern, and select the candidate with minimum truncated objective. Then, we evaluate the selected subspace on all projected columns in R and remove those reaching the threshold. The truncated objective simultaneously controls the number of thresholded columns and the reconstruction error of the retained columns.

Finally, let T denote the columns retained after thresholding. We show that at most $M = (1+\varepsilon)m$ original columns lie outside T and that $\|A_T - (A_T)_k\|_F^2 = O(f^\star/\varepsilon)$. We then invoke a standard $O(k)$-approximate exact-k CSS routine VOLUMECSS to select exactly k original columns. As in the bicriteria algorithm, the intermediate removals are used only to construct the retained instance; the final discarded set is determined by residual trimming on the original matrix.

We now analyze these stages. Let $\Delta = \lceil \varepsilon m \rceil$ and $K = C_0 k/\varepsilon$, where $C_0 \ge 128$ is an absolute constant. By Theorem B.1, with probability at least $7/8$, the exact-score peeling stage of Algorithm 3 completes without reaching its removal limit and returns $R = [d] \setminus D$ satisfying $Z_\lambda(A_R) \le K$ and $|D \cap I^\star| \le \Delta/4$. Let $V \in \mathbb{R}^{n\times r}$ contain the left singular vectors of A_R whose squared singular values exceed λ, and set $z_j = V^\top a_j$ for $j \in R$.

Lemma 4.1. *The ridge-mass bound implies that $r \le 2K = O(k/\varepsilon)$. The projection onto V also satisfies $\|(I_n - VV^\top)A_R\|_F^2 \le 2\lambda K = O(f^\star/\varepsilon)$.*

Thus, the remaining robust search takes place in dimension $O(k/\varepsilon)$, with projection error already within the target scale. Let $\theta = 64\widehat{f}/\Delta$. For any subspace $W \subseteq \mathbb{R}^r$ with $\dim(W) \le k$, define

$$\Phi_\theta(W) = \sum_{j \in R} \min\left\{\|(I_r - \Pi_W)z_j\|_2^2, \theta\right\}.$$

Let $q_R = |N^\star \cap R|$, and let W_0 be the projection of the optimal comparison subspace onto $\mathrm{span}(V)$. The optimal inliers contribute at most $f^\star$, while each remaining optimal outlier contributes at most θ. Hence, we have $\min_{\dim(W)\le k} \Phi_\theta(W) \le \Phi_\theta(W_0) \le f^\star + q_R\theta$. The term $q_R\theta$ accounts for the remaining optimal outliers, while the excess controls the additional columns removed by thresholding.

Although the projected dimension is $O(k/\varepsilon)$, directly enumerating threshold patterns over all columns in R would require $|R|^{\mathrm{poly}(k,1/\varepsilon)}$ time and is therefore not fixed-parameter tractable. We first compress the projected instance. With $\xi = \varepsilon/256$, we construct a strong coreset satisfying

$$(1-\xi)\Phi_\theta(W) \le \widetilde{\Phi}_\theta(W) \le (1+\xi)\Phi_\theta(W)$$

for every subspace W of dimension at most k.

The construction lifts squared projection residuals to dimension $p = O(k^2/\varepsilon^2)$. Since each residual is affine in the corresponding orthogonal projector, repeated basis selection yields sensitivity upper bounds, from which sensitivity sampling constructs a weighted coreset of size $2^{O(p)}\,\mathrm{poly}(k, 1/\varepsilon)\,\mathrm{polylog}(d+2)$. The construction and analysis are deferred to the appendix.

We solve the coreset by enumerating all realizable residual-threshold patterns. For each pattern, we compute the weighted PCA minimizer on its below-threshold columns and evaluate the resulting subspace under the full truncated coreset objective. The resulting candidate need not realize the pattern that generated it. Exactness follows because the pattern induced by an optimal coreset solution is enumerated, and replacing its subspace by the corresponding weighted PCA minimizer cannot increase the truncated objective. The enumeration procedure is given in the appendix.

Lemma 4.2. *Conditioned on successful peeling, the coreset construction and exact search return, with constant probability, a subspace $\widehat{W}$ of dimension at most k satisfying $\Phi_\theta(\widehat{W}) \le q_R\theta + 4f^\star$. The running time is $2^{O(k^4/\varepsilon^4)}\,\mathrm{poly}(n,d)$.*

The coreset accuracy ensures that the excess beyond $q_R\theta$ is only $O(f^\star)$. Let $B = \{j \in R : \|(I_r - \Pi_{\widehat{W}})z_j\|_2^2 \ge \theta\}$ and $T = R \setminus B$. Then, we have $\Phi_\theta(\widehat{W}) = \theta|B| + \sum_{j\in T}\|(I_r - \Pi_{\widehat{W}})z_j\|_2^2$. The first term controls the number of thresholded columns, while the second is the projected reconstruction error of the retained columns.

Lemma 4.3. *On the successful events above, $|D \cup B| \le M$ and $\|A_T - (A_T)_k\|_F^2 = O(f^\star/\varepsilon)$.*

It remains to select exactly k original columns. If $|T| \ge k$, we apply the standard $O(k)$-approximate VOLUMECSS routine to A_T. Together with Lemma 4.3, this gives $\|(I_n - \Pi_{A_S})A_T\|_F^2 = O(k/\varepsilon)f^\star$. If $|T| < k$, we select all columns in T and add $k - |T|$ other original columns, giving zero reconstruction error on A_T. The sets D and B are used only to construct the retained instance. For comparison, let $\widetilde{N} = (D \cup B) \setminus S$. By Lemma 4.3, $|\widetilde{N}| \le M$ and $\widetilde{N} \cap S = \emptyset$. After removing $\widetilde{N}$, every remaining column belongs to $T \cup S$, and the columns in S have zero residual. Hence, the remaining error equals the residual error on A_T. The algorithm instead removes the $\min\{M, d-k\}$ largest residuals outside S on the original matrix, which cannot have larger reconstruction error.

Theorem 4.4. *Suppose $1 \le k \le d$, $m \ge 1$, $0 < \varepsilon \le 1/2$, $f^\star > 0$, and $f^\star \le \widehat{f} \le 2f^\star$. With constant probability, Algorithm 3 returns disjoint sets $S, N \subseteq [d]$ satisfying $|S| = k$ and $|N| \le (1+\varepsilon)m$, with $\|(I_n - \Pi_{A_S})A_{[d]\setminus N}\|_F^2 \le O(k/\varepsilon)f^\star$. The algorithm runs in $O\left(2^{k^4/\varepsilon^4}d^2 + m(n^3 + n^2 d) + kdn^\omega \log n\right)$, where ω is the matrix multiplication exponent.*

5 CONCLUSION

We develop two approximation algorithms for CSS with outliers while discarding at most $(1+\varepsilon)m$ columns. Our bicriteria algorithm achieves a $(1+\varepsilon)$-approximation using $O(k/\varepsilon^2)$ selected columns, removing the dependence on d from the selected-column bound. For exact-k selection, we give a fixed-parameter tractable algorithm with an $O(k/\varepsilon)$-approximation. Improving the exact-k approximation factor and removing the relaxation in the outlier budget remain open directions.

AI USE STATEMENT

We used generative AI tools to assist with the revision and presentation of the manuscript. In particular, AI tools were used to help check the consistency of notation and definitions, identify potential issues in proof exposition, and improve the writing of the text. All AI-assisted mathematical arguments were reviewed and verified by the authors. The authors take full responsibility for the correctness, originality, and final content of the paper.

ETHICS STATEMENT

This work is theoretical and studies algorithms for column subset selection with outliers. It does not involve human subjects, personal or sensitive data, user studies, or the collection or release of datasets. The proposed methods operate on abstract matrix inputs and are intended as general-purpose tools for dimensionality reduction and feature selection. We are not aware of ethical concerns specific to the theoretical results presented in this paper.

REPRODUCIBILITY STATEMENT

This paper is theoretical. The problem formulation, assumptions, and parameter choices are specified in Sections 2–4, and the proposed algorithms are given explicitly in pseudocode. Appendices A and B contain the complete proofs of the main results and supporting lemmas, as well as implementation details, probability guarantees, and running-time analyses of the relevant subroutines. No experimental datasets or empirical results are used in this work, and all information needed to verify the stated theoretical guarantees is contained in the paper and its appendices.

REFERENCES

Thomas Abeel, Thibault Helleputte, Yves Van de Peer, Pierre Dupont, and Yvan Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, 2010.

Joshua Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

Aditya Bhaskara and Srivatsan Kumar. Low rank approximation in the presence of outliers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 4:1–4:16, 2018.

Christos Boutsidis, Michael W Mahoney, and Petros Drineas. Unsupervised feature selection for principal components analysis. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pp. 61–69, 2008.

Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 968–977, 2009a.

Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. Unsupervised feature selection for the k-means clustering problem. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, pp. 153–161, 2009b.

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014.

Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pp. 181–190, 2015.

Michael B Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1758–1777, 2017.

Yogesh Dahiya, Fedor Fomin, Fahad Panolan, and Kirill Simonov. Fixed-parameter and approximation algorithms for PCA with outliers. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2341–2351, 2021.

Amit Deshpande and Rameshwar Pratap. Sampling-based dimension reduction for subspace approximation with outliers. *Theoretical Computer Science*, 858:100–113, 2021.

Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *Proceedings of the 51st IEEE Annual Symposium on Foundations of Computer Science*, pp. 329–338, 2010.

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1117–1126, 2006.

Petros Drineas, Michael W Mahoney, and Shan Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

pages=5818–5826 year=2019 Fomin, Fedor and Golovach, Petr and Panolan, Fahad and Simonov, Kirill, booktitle=Proceedings of the 36th International Conference on Machine Learning. Refined complexity of PCA with outliers.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

Sajjad Hashemian, Mohammad Saeed Arvenaghi, and Ebrahim Ardeshir-Larijani. Optimal bound for PCA with outliers using higher-degree voronoi diagrams. *Computational Geometry*, 132:102231, 2026.

Shuli Jiang, Dennis Li, Irene Mengze Li, Arvind V Mahankali, and David Woodruff. Streaming and distributed algorithms for robust column subset selection. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 4971–4981, 2021.

Arvind V Mahankali and David P Woodruff. Optimal ℓ_1 column subset selection and a fast ptas for low rank approximation. In *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms*, pp. 560–578, 2021.

Michael W Mahoney and Petros Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 688–701, 2017.

Zhao Song, David Woodruff, and Peilin Zhong. Average case column subset selection for entrywise ℓ_1-norm loss. In *Proceedings of the 33th International Conference on Neural Information Processing Systems*, pp. 10111–10121, 2019.

David P Woodruff and Taisuke Yasuda. New subset selection algorithms for low rank approximation: Offline and online. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pp. 1802–1813, 2023.

Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, volume 23, pp. 2496–2504, 2010.

Tianhao Zhu and Jie Shen. Residual-based sampling for online outlier-robust PCA. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 27591–27611, 2022.

A Proofs and Implementation for Bicriteria Approximation

This section provides the proofs and implementation details for RidgePeel and RobustCSS. We first establish the ridge-score bounds and the low-dimensional representation used within a peeling call. Then, we describe the auxiliary tree for dynamic score sampling, analyze the hierarchical peeling process, and complete the column-selection and residual-trimming guarantees. The outer bucket hierarchy and the auxiliary tree used within each peeling call serve different purposes and are analyzed separately.

A.1 Notation and Basic Ridge-Score Bounds

We use the notation from the main text. Let $N^\star$ be a fixed optimal outlier set, let $I^\star = [d] \setminus N^\star$, and let $U^\star$ be a best rank-k approximation subspace for $A_{I^\star}$, with $P^\star$ denoting the orthogonal projector onto $U^\star$. Then $f^\star = \|(I_n - P^\star)A_{I^\star}\|_F^2 > 0$. Given $f^\star \le \widehat{f} \le 2f^\star$ and $\lambda = \widehat{f}/k$, we set $\Delta = \lceil \varepsilon m \rceil$, $M = m + \Delta$, and $K = 7200k/\varepsilon$.

For any active set $R \subseteq [d]$, let $\tau_j^R = a_j^\top (A_R A_R^\top + \lambda I_n)^{-1} a_j$ denote the ridge leverage score of column j, and let $Z_\lambda(A_R) = \sum_{j \in R} \tau_j^R$ denote the total ridge leverage mass. All calls to RidgePeel within one execution of RobustCSS share the same removal set D. This set records intermediate peeling removals and is distinct from the final discarded set N. For $m \ge 1$, we have $\Delta \le m$ and $M \le 2m$; when $m = 0$, the peeling stage is skipped.

The following ridge-score properties are slight extensions of standard ridge leverage arguments from Cohen et al. (2017) to the surviving optimal-inlier sets in our outlier setting.

Lemma A.1. *For every active set $R \subseteq [d]$, the surviving optimal inliers contribute at most $2k$ total ridge leverage mass $\sum_{j \in R \cap I^\star} \tau_j^R \le 2k$.*

Proof. Let $G = R \cap I^\star$. Since $A_R A_R^\top \succeq A_G A_G^\top$, inverse monotonicity implies that the ridge leverage scores of columns in G can only decrease when the remaining columns of R are added. Hence, their total mass is at most $Z_\lambda(A_G)$. Since $G \subseteq I^\star$ and $U^\star$ has dimension at most k, $U^\star$ is a feasible rank-k subspace for A_G. Therefore, we have

$$\|A_G - (A_G)_k\|_F^2 \le \|(I_n - P^\star)A_G\|_F^2 \le \|(I_n - P^\star)A_{I^\star}\|_F^2 = f^\star.$$

Thus, it holds that

$$\sum_{j \in G} \tau_j^R \le Z_\lambda(A_G) = \sum_i \frac{\sigma_i(A_G)^2}{\sigma_i(A_G)^2 + \lambda} \le k + \frac{\|A_G - (A_G)_k\|_F^2}{\lambda} \le k + \frac{f^\star}{\lambda} \le 2k,$$

where the last inequality follows from $\lambda = \widehat{f}/k$ and $f^\star \le \widehat{f}$. □

Lemma A.2. *Every active column has ridge leverage score at most one. In particular, for every $R \subseteq [d]$ and $j \in R$, the score satisfies $\tau_j^R \le 1$.*

Proof. The ridge matrix satisfies $A_R A_R^\top + \lambda I_n \succeq a_j a_j^\top + \lambda I_n$. Therefore, using the inverse monotonicity and the Sherman–Morrison identity gives

$$\tau_j^R \le a_j^\top (a_j a_j^\top + \lambda I_n)^{-1} a_j = \frac{\|a_j\|_2^2}{\|a_j\|_2^2 + \lambda} \le 1.$$

□

We also use the monotonicity of ridge leverage scores under column addition from Cohen et al. (2017). Applying this property to disjoint column blocks gives the following subadditivity bound.

Lemma A.3. *For disjoint sets $R_1, \dots, R_s$ with $R = \cup_{i=1}^s R_i$, the total ridge leverage mass satisfies*

$$Z_\lambda(A_R) \le \sum_{i=1}^s Z_\lambda(A_{R_i}).$$

Moreover, removing columns cannot increase the total ridge leverage mass, so $Z_\lambda(A_{R'}) \le Z_\lambda(A_R)$ whenever $R' \subseteq R$.

Proof. For every $j \in R_i$, adding the columns outside R_i increases the ridge matrix and therefore decreases the score of column j. Thus $\tau_j^R \le \tau_j^{R_i}$, and summing over the disjoint blocks gives the subadditivity bound.

For monotonicity, let $B' = A_{R'}A_{R'}^\top$ and $B = A_R A_R^\top$. Since $B' \preceq B$, inverse monotonicity gives

$$I_n - \lambda(B' + \lambda I_n)^{-1} \preceq I_n - \lambda(B + \lambda I_n)^{-1}.$$

Taking traces yields $Z_\lambda(A_{R'}) \le Z_\lambda(A_R)$. □

A.2 Ridge Sampling and Deletion-Stable Score Estimates

Since the surviving set changes after every deletion, RidgePeel requires score estimates that remain valid throughout the entire call. We achieve this by constructing a low-dimensional representation once during preprocessing and reusing it after subsequent deletions. We first describe the ridge-score estimation routine used to build this representation, and then prove that the resulting estimates remain valid for every surviving subset generated by the peeling process.

Fast ridge-score estimation. We adapt the recursive column-sampling construction of Cohen et al. (2017) to an externally specified regularization parameter λ. We refer to the resulting routine as FastApproxRidgeScores. This routine is not a new algorithm introduced in our work. Given a matrix B, a ridge parameter λ, an upper bound on its total ridge leverage mass, and a failure probability, it returns constant-factor upper estimates of the ridge leverage scores of all columns.

Lemma A.4 (Adapted from Cohen et al. (2017))**.** *Let $B = [b_1, \dots, b_{d_B}] \in \mathbb{R}^{n \times d_B}$ satisfy $Z_\lambda(B) \le H$, where $\lambda > 0$ and $H \ge 1$. With probability at least $1 - \delta$, $\mathrm{FastApproxRidgeScores}(B, \lambda, H, \delta)$ returns estimates $u_1, \dots, u_{d_B}$ satisfying*

$$\tau_j^\lambda(B) \le u_j \le c_{\mathrm{lev}} \tau_j^\lambda(B)$$

for every $j \in [d_B]$, where $\tau_j^\lambda(B) = b_j^\top (BB^\top + \lambda I_n)^{-1} b_j$ and c_{lev} is an absolute constant. Thus $\sum_j u_j \le c_{\mathrm{lev}} H$. The routine can be implemented in $\widetilde{O}\left(\mathrm{nnz}(B)H + (n + d_B)H^2 + H^3\right)$ time.

Proof. We describe an implementation for the prescribed regularization parameter λ. If $B = 0$, return $u_j = 0$ for every j. Otherwise, set $q = \lceil 2H \rceil$. If $\min\{n, d_B\} \le q$, a thin SVD of B allows us to compute all scores exactly in $O((n + d_B)q^2)$ time, which is within the claimed bound. Hence, assume that $q < \min\{n, d_B\}$.

We first relate the prescribed scale λ to the rank-q tail of B. Since $\sum_i \frac{\sigma_i(B)^2}{\sigma_i(B)^2 + \lambda} = Z_\lambda(B) \le H$, we have $\sigma_i(B)^2 \le \lambda$ for every $i > q$. Indeed, otherwise the first $q + 1$ singular directions would each contribute more than $1/2$ to the sum, contradicting $(q + 1)/2 > H$. Consequently, we have

$$\|B - B_q\|_F^2 = \sum_{i>q} \sigma_i(B)^2 \le 2\lambda \sum_{i>q} \frac{\sigma_i(B)^2}{\sigma_i(B)^2 + \lambda} \le 2\lambda H \le q\lambda.$$

Thus, the auxiliary scale $\lambda_q := \frac{\|B - B_q\|_F^2}{q}$ satisfies $\lambda_q \le \lambda$. This quantity is used only in the analysis. The routine does not need to compute B_q or λ_q.

We apply the recursive column-sampling construction of Cohen et al. (2017) with target rank q, constant accuracy $1/4$, and total failure probability at most δ. Its additive-multiplicative spectral guarantee gives a reweighted column sample $\widehat{B} \in \mathbb{R}^{n \times s}$, with $s = \widetilde{O}(q)$, such that

$$\frac{3}{4}\widehat{B}\widehat{B}^\top - \frac{1}{4}\lambda_q I_n \preceq BB^\top \preceq \frac{5}{4}\widehat{B}\widehat{B}^\top + \frac{1}{4}\lambda_q I_n.$$

The zero-tail case $\lambda_q = 0$ is included using the pseudoinverse convention in that construction.

Let $G = BB^\top + \lambda I_n$ and $\widehat{G} = \widehat{B}\widehat{B}^\top + \lambda I_n$. Since $\lambda_q \le \lambda$, the preceding inequalities imply $\frac{3}{4}\widehat{G} \preceq G \preceq \frac{5}{4}\widehat{G}$. Therefore, we get $\frac{4}{5}G \preceq \widehat{G} \preceq \frac{4}{3}G$. In particular, setting $\alpha = 1/3$, we obtain

$$(1 - \alpha)(BB^\top + \lambda I_n) \preceq \widehat{B}\widehat{B}^\top + \lambda I_n \preceq (1 + \alpha)(BB^\top + \lambda I_n). \tag{2}$$

Define $u_j = (1+\alpha) b_j^\top \widehat{G}^{-1} b_j$. Both G and $\widehat{G}$ are positive definite since $\lambda > 0$. Applying the inverse monotonicity to Equation (2) gives

$$\tau_j^\lambda(B) \le u_j \le \frac{1+\alpha}{1-\alpha} \tau_j^\lambda(B) = 2\tau_j^\lambda(B).$$

Thus, we may take $c_{\mathrm{lev}} = 2$, and $\sum_j u_j \le 2Z_\lambda(B) \le 2H$.

It remains to bound the running time. The recursive sample can be constructed in $\widetilde{O}(\mathrm{nnz}(B) + nq^2 + d_B)$ time using the basic recursive implementation of Cohen et al. (2017). To evaluate the quadratic forms, compute a compact SVD for $\widehat{B} = U\Sigma V^\top$ with $\Sigma = \mathrm{diag}(\gamma_1, \dots, \gamma_\ell)$, where $\ell \le s$ and the columns of U are orthonormal. For $x_j = U^\top b_j$, we have

$$b_j^\top \widehat{G}^{-1} b_j = \sum_{i=1}^{\ell} \frac{(x_j)_i^2}{\gamma_i^2 + \lambda} + \frac{\|b_j\|_2^2 - \|x_j\|_2^2}{\lambda}.$$

Thus, no $n \times n$ inverse needs to be formed. The SVD costs $O(ns^2 + s^3)$ time, while computing $U^\top B$, the column norms, and all quadratic forms costs $O(\mathrm{nnz}(B)s + d_B s + d_B)$ time. Using $q = O(H)$ and $s = \widetilde{O}(H)$, the total running time is bounded by $\widetilde{O}\left(\mathrm{nnz}(B)H + (n + d_B)H^2 + H^3\right)$. Here, all computations after constructing $\widehat{B}$ are deterministic, so the simultaneous score guarantees hold with probability at least $1 - \delta$. □

Constructing a fixed representation for one block. we consider one call to RIDGEPEEL with input block U and a valid bound $Z_\lambda(A_U) \le H$. We use the preceding ridge-score estimates in constant-accuracy ridge leverage sampling to construct a reweighted sample C. With probability at least $1 - \eta$, the sample contains $O(H \log((H+1)/\eta))$ columns and satisfies

$$\frac{1}{2}(A_U A_U^\top + \lambda I_n) \preceq CC^\top + \lambda I_n \preceq \frac{3}{2}(A_U A_U^\top + \lambda I_n). \tag{3}$$

The sampled columns in C are used only to construct the low-dimensional representation, while the peeling procedure continues to operate on the full active set U.

Let $V \in \mathbb{R}^{n \times s}$ be an orthonormal basis for $\mathrm{col}(C)$. The sample-size bound gives $s = O(H \log((H+1)/\eta))$. More importantly, the sampled subspace also controls the residual part of the entire block. For any vector z orthogonal to $\mathrm{col}(V)$, we have $CC^\top z = 0$. Therefore, applying the lower inequality in Equation (3) gives

$$\frac{1}{2}\left(z^\top A_U A_U^\top z + \lambda \|z\|_2^2\right) \le \lambda \|z\|_2^2.$$

Hence, the component of A_U outside the sampled subspace satisfies

$$\|(I_n - VV^\top) A_U\|_2^2 \le \lambda. \tag{4}$$

We fix this matrix V throughout the entire peeling call. For each $j \in U$, let $x_j = V^\top a_j$ denote the coordinates of a_j inside the sampled subspace. We write $e_j = (I_n - VV^\top)a_j$ for its residual component and define $t_j = \|e_j\|_2^2/\lambda$. Thus, x_j records the contribution inside the low-dimensional representation, while t_j records the residual contribution outside this representation. All three quantities remain unchanged as other columns are removed.

For a current surviving set $R \subseteq U$, let $X_R = [x_j]_{j \in R}$ and define $P_R = (X_R X_R^\top + \lambda I_s)^{-1}$. The score estimate maintained by RIDGEPEEL is

$$q_j^R = 3\left(x_j^\top P_R x_j + t_j\right).$$

The representation V and the column quantities x_j, t_j remain fixed, while P_R changes with the current surviving set. The next lemma shows that these fixed quantities continue to provide constant-factor estimates of the current ridge leverage scores after arbitrary deletions.

Lemma A.5. *Suppose Equation* (4) *holds. For every surviving set $R \subseteq U$ and every $j \in R$, the maintained estimate satisfies*

$$\tau_j^R \le q_j^R \le 9\tau_j^R. \tag{5}$$

This guarantee holds simultaneously for all $R \subseteq U$, including sets chosen adaptively by previous deletions. If $Q_R = \sum_{j \in R} q_j^R$, then $Z_\lambda(A_R) \le Q_R \le 9Z_\lambda(A_R)$. Moreover, the surviving optimal inliers contribute at most $18k$ estimated ridge leverage mass.

Proof. The proof separates the contribution inside $\mathrm{col}(V)$ from the residual outside this subspace. The ridge variational identity gives

$$\tau_j^R = \min_{c \in \mathbb{R}^{|R|}} \left\{ \|c\|_2^2 + \lambda^{-1} \|a_j - A_R c\|_2^2 \right\}.$$

Let $E_R = [e_i]_{i \in R}$. Since the columns of V are orthogonal to the residuals in E_R, the same objective can be written as

$$\tau_j^R = \min_c \left\{ \|c\|_2^2 + \lambda^{-1} \|x_j - X_R c\|_2^2 + \lambda^{-1} \|e_j - E_R c\|_2^2 \right\}.$$

The ridge objective restricted to the low-dimensional coordinates has value

$$u_j^R = x_j^\top P_R x_j = \min_c \left\{ \|c\|_2^2 + \lambda^{-1} \|x_j - X_R c\|_2^2 \right\}.$$

Dropping the nonnegative residual term immediately gives $u_j^R \le \tau_j^R$.

We next bound the residual contribution t_j. Let $c_\star$ minimize the full variational expression. Since $R \subseteq U$, Equation (4) implies $\|E_R\|_2^2 \le \lambda$. Therefore, we have

$$t_j = \lambda^{-1} \|e_j\|_2^2 \le 2\lambda^{-1} \|e_j - E_R c_\star\|_2^2 + 2\lambda^{-1} \|E_R c_\star\|_2^2 \le 2\tau_j^R.$$

Combining with $u_j^R \le \tau_j^R$, this gives $u_j^R + t_j \le 3\tau_j^R$.

For the reverse comparison, let c_x minimize the low-dimensional objective defining u_j^R. Evaluating the full ridge objective at c_x gives

$$\begin{aligned} \tau_j^R &\le u_j^R + \lambda^{-1} \|e_j - E_R c_x\|_2^2 \\ &\le u_j^R + 2t_j + 2\|c_x\|_2^2 \\ &\le 3u_j^R + 2t_j \\ &\le 3(u_j^R + t_j). \end{aligned}$$

Since $q_j^R = 3(u_j^R + t_j)$, the two bounds imply $\tau_j^R \le q_j^R \le 9\tau_j^R$.

Summing the pointwise estimates gives $Z_\lambda(A_R) \le Q_R \le 9Z_\lambda(A_R)$. Applying Lemma A.1 to the upper pointwise bound also gives

$$\sum_{j \in R \cap I^\star} q_j^R \le 18k.$$

The argument is deterministic once V is fixed and holds for every subset $R \subseteq U$. Therefore, the same guarantee applies to the adaptively generated surviving sets without a union bound over those sets. □

A.3 Auxiliary Tree for Dynamic Score Sampling

In this section, we introduce an auxiliary tree structure to implement the score-proportional sampling and deletion steps in Algorithm 1 without rescanning all surviving columns after each deletion. Consider one nontrivial call to RidgePeel with input block U. The low-dimensional basis V and the quantities x_j, t_j for $j \in U$ are fixed throughout this call. For a current surviving set $R \subseteq U$, however, the matrix $P_R = (X_R X_R^\top + \lambda I_s)^{-1}$ changes whenever a column is removed. Hence the estimated score $q_j^R = 3(x_j^\top P_R x_j + t_j)$ of every surviving column may also change after each deletion.

The key idea is to maintain the total score of a group of columns without recomputing the score of every column in that group. For any subset $W \subseteq R$, define $G_W = \sum_{j \in W} x_j x_j^\top$ to aggregate the contribution of these columns inside the low-dimensional representation, and define $T_W = \sum_{j \in W} t_j$ to aggregate their residual contribution outside this representation. Their total unnormalized score can then be evaluated as

$$\sum_{j \in W} \left(x_j^\top P_R x_j + t_j \right) = \mathrm{tr}(P_R G_W) + T_W.$$

Thus, once G_W and T_W are stored, the current score of the entire group can be obtained from the latest P_R without visiting its individual columns. We organize these aggregate statistics in a balanced binary tree and use them to locate one sampled column hierarchically.

Tree structure. Fix an arbitrary ordering of the indices in U. We recursively split this ordering into two parts whose sizes differ by at most one, producing a balanced binary tree $\mathcal{T}_U$ with one leaf for each original index in U. The tree has $O(|U|)$ nodes and height $O(\log(|U|+2))$. We also store a pointer from each original index to its leaf. The topology of $\mathcal{T}_U$ remains fixed throughout the entire RIDGEPEEL call.

For a node v, let $U_v \subseteq U$ denote the fixed set of leaf indices in its subtree. Given the current surviving set R, let $R_v = U_v \cap R$ denote the active indices below v. Each node stores two aggregate statistics. The matrix $G_v = \sum_{j\in R_v} x_j x_j^\top$ records the low-dimensional contribution of the active columns below the node, while $T_v = \sum_{j\in R_v} t_j$ records their residual contribution. At an active leaf corresponding to column j, we have $G_v = x_j x_j^\top$ and $T_v = t_j$. Internal statistics are obtained by adding the corresponding statistics of the two children, so the tree can be initialized bottom-up.

Deleting a column does not remove its leaf or change the tree topology. Instead, the leaf is marked inactive and its two contributions are set to zero. This allows every original column to retain a fixed leaf location throughout the call and avoids any tree rebalancing.

There is a single matrix P_R shared by all nodes of the auxiliary tree. Given the current P_R, the weight of node v is defined by

$$w_v(P_R) = \mathrm{tr}(P_R G_v) + T_v.$$

By the definitions of G_v and T_v, this weight is exactly the total unnormalized estimated score of the active columns in the subtree

$$w_v(P_R) = \sum_{j\in R_v} \left(x_j^\top P_R x_j + t_j\right).$$

If o denotes the root, then $G_o = \sum_{j\in R} x_j x_j^\top$, $T_o = \sum_{j\in R} t_j$, and $w_o(P_R) = Q_R/3$.

The tree stores only the aggregate statistics G_v and T_v for each node. When a node is visited during sampling, its current weight is computed from these statistics using the current matrix P_R. After a deletion, P_R changes, so the weight of every subtree may also change, including subtrees that do not contain the deleted column. However, their stored statistics remain unchanged. Therefore, only the moments along the deleted leaf-to-root path need to be updated, while all other node weights are recomputed only when they are next used.

Sampling from the current estimated scores. The tree is used to sample exactly from the distribution specified by RIDGEPEEL. Starting at the root, suppose the current node v has children v_1 and v_2. We evaluate their current weights $w_{v_1}(P_R)$ and $w_{v_2}(P_R)$ and descend to v_1 with probability $\frac{w_{v_1}(P_R)}{w_{v_1}(P_R)+w_{v_2}(P_R)}$, and to v_2 otherwise. Repeating this step reaches a leaf after $O(\log(|U|+2))$ levels.

All node weights are nonnegative because $P_R \succ 0$, $G_v \succeq 0$, and $T_v \geq 0$. A child with zero weight is assigned probability zero. Sampling is called only when $Q_R > 0$, so every visited internal node has positive total weight and no zero denominator occurs.

Lemma A.6. *Conditioned on the current surviving set R and matrix P_R, the auxiliary-tree descent returns each $j \in R$ with probability* $\Pr[J=j] = \frac{q_j^R}{Q_R}$.

Proof. Let $o = v_0, v_1, \ldots, v_h = v_j$ be the unique root-to-leaf path ending at the leaf corresponding to j. Since the moment statistics are additive, the weight of every internal node equals the sum of the weights of its two children. Therefore, the conditional probability of moving from $v_{\ell-1}$ to v_ℓ is $w_{v_\ell}(P_R)/w_{v_{\ell-1}}(P_R)$. Multiplying these conditional probabilities gives

$$\Pr[J=j] = \prod_{\ell=1}^{h} \frac{w_{v_\ell}(P_R)}{w_{v_{\ell-1}}(P_R)} = \frac{w_{v_j}(P_R)}{w_o(P_R)} = \frac{x_j^\top P_R x_j + t_j}{\sum_{i\in R}(x_i^\top P_R x_i + t_i)} = \frac{q_j^R}{Q_R}.$$

A zero-weight leaf is never selected and has target probability zero. For a tree containing a single active leaf, the same conclusion holds trivially. The tree therefore implements exactly the distribution induced by the estimated scores q_j^R. No claim is made that this distribution is proportional to the exact ridge leverage scores τ_j^R. □

Algorithm 4 AUXILIARYSAMPLEANDDELETE

Input: current inverse P, auxiliary tree with moments $\{G_v, T_v\}$, surviving set R, and shared removal set D.
Output: sampled index J and the updated state.

1: Set v to the root.
2: **while** v is not a leaf **do**
3: Let v_1, v_2 be the children of v.
4: Compute $w_1 \leftarrow \mathrm{tr}(PG_{v_1}) + T_{v_1}$ and $w_2 \leftarrow \mathrm{tr}(PG_{v_2}) + T_{v_2}$.
5: Set $v \leftarrow v_1$ with probability $w_1/(w_1 + w_2)$, and set $v \leftarrow v_2$ otherwise.
6: **end while**
7: Let J be the original index stored at leaf v.
8: Set $y \leftarrow Px_J$ and $P \leftarrow P + yy^\top/(1 - x_J^\top y)$.
9: **for** nodes u on the path from v to the root **do**
10: Set $G_u \leftarrow G_u - x_J x_J^\top$ and $T_u \leftarrow T_u - t_J$.
11: **end for**
12: Mark leaf v inactive, set $R \leftarrow R \setminus \{J\}$, and set $D \leftarrow D \cup \{J\}$.
13: Set $Q \leftarrow 3(\mathrm{tr}(PG_o) + T_o)$, where o is the root.
14: **return** J and the updated P, R, D, Q and tree.

Updating after a deletion. Suppose the sampled index is J. Removing this column changes the coordinate Gram matrix from $X_R X_R^\top$ to $X_R X_R^\top - x_J x_J^\top$. Recomputing the inverse from scratch would be unnecessary. Let $y = P_R x_J$. The rank-one downdate gives

$$P_{R\setminus\{J\}} = P_R + \frac{yy^\top}{1 - x_J^\top y}. \tag{6}$$

The denominator in Equation (6) is positive. Indeed, let $C = X_R X_R^\top + \lambda I_s$ and $C' = X_{R\setminus\{J\}} X_{R\setminus\{J\}}^\top + \lambda I_s$. Both matrices are positive definite and satisfy $C' = C - x_J x_J^\top$. The matrix determinant lemma gives $\det(C') = \det(C)(1 - x_J^\top C^{-1} x_J)$, which implies $1 - x_J^\top P_R x_J > 0$. Sherman–Morrison then yields Equation (6).

The stored moments change only for nodes whose subtrees contain J. These are exactly the nodes on the path from the leaf of J to the root. Along this path, we update $G_v \leftarrow G_v - x_J x_J^\top$ and $T_v \leftarrow T_v - t_J$. The leaf is then marked inactive, J is removed from R, and the same index is added to the shared removal set D.

No other node moment changes. This does not mean that the weights of the other nodes remain unchanged: the update of P_R may change all current weights. Their stored data are nevertheless still correct, and their new weights are obtained automatically when they are next evaluated using $P_{R\setminus\{J\}}$. Hence the deletion requires no scan over the remaining leaves and no refresh of all individual scores.

Lemma A.7. *After applying Equation* (6) *and updating the moments along the leaf-to-root path of J, every node v again satisfies $G_v = \sum_{j\in R_v} x_j x_j^\top$ and $T_v = \sum_{j\in R_v} t_j$ for the new surviving set. Consequently, every subsequently evaluated node weight equals the current aggregate score of its active subtree.*

Proof. The deleted index J belongs to U_v exactly for the nodes on its leaf-to-root path. Subtracting $x_J x_J^\top$ and t_J at precisely these nodes therefore removes its contribution from every affected subtree and leaves all other subtree moments unchanged. The inverse update gives the exact matrix $(X_{R\setminus\{J\}} X_{R\setminus\{J\}}^\top + \lambda I_s)^{-1}$. Thus the stored moments and the shared inverse represent the new surviving set exactly. Repeating the same argument after every deletion preserves the invariant throughout the call. □

The complete sampling and deletion operation can therefore be implemented as follows.

Running time and boundary cases. The auxiliary tree has $O(|U|)$ nodes and depth $O(\log(|U|+2))$. Each node stores one $s \times s$ matrix and one scalar. Once the fixed coordinates x_j and residual terms

t_j are available, initializing the leaf moments and summing them bottom-up requires $O(|U|(s^2+1))$ arithmetic operations and storage.

Evaluating one node weight requires computing $\mathrm{tr}(P_R G_v)$ and therefore costs $O(s^2+1)$ arithmetic operations. Sampling follows one root-to-leaf path, and deletion updates the same number of nodes. The rank-one inverse update and the final root-weight evaluation each require $O(s^2+1)$ additional work.

Lemma A.8. *Given the fixed coordinates x_j and residual terms t_j, building the auxiliary tree requires $O(|U|(s^2+1))$ arithmetic operations and storage. One exact estimated-score sample, deletion, and root-score refresh require $O((s^2+1)\log(|U|+2))$ time.*

Proof. The initialization bound follows from the $O(|U|)$ tree nodes and the $s \times s$ matrix stored at each node. During sampling, one node is visited at each tree level and each child weight costs $O(s^2+1)$ to evaluate. Deletion modifies one leaf-to-root path of the same length. The inverse update and root-weight refresh do not change the asymptotic bound. □

The coordinate array $\{x_j\}_{j\in U}$ requires $O(|U|s)$ additional storage, and storing the basis V requires $O(ns)$ storage. These costs are separate from the auxiliary-tree storage. If RIDGEPEEL returns immediately because U is empty, $A_U = 0$, or the supplied input bound already satisfies $H \le L$, no auxiliary tree is constructed. When $s = 0$, the matrix moments are empty and the node weights depend only on the scalar residual terms t_j; the same tree construction applies.

Finally, the auxiliary tree is local to a single RIDGEPEEL call. Once that call finishes, its tree, coordinates, and local basis can be released. The outer hierarchy in ROBUSTCSS passes only surviving original column indices to the parent node. A parent that requires preprocessing constructs a new basis and a new auxiliary tree for its own merged block. Therefore, the outer bucket hierarchy determines which groups of original columns are processed together, while the auxiliary tree only implements dynamic sampling and deletions within one such block.

A.4 HIERARCHICAL PEELING AND GLOBAL REMOVAL ACCOUNTING

We now analyze the outer hierarchy used by ROBUSTCSS. This hierarchy is different from the auxiliary tree in Appendix A.3. The auxiliary tree accelerates score-proportional sampling within one RIDGEPEEL call, while the outer hierarchy determines which column blocks are passed to different peeling calls. Its purpose is to keep the input ridge leverage mass of every local call controlled without processing the full matrix with the initial bound $2k+m$.

For one execution of ROBUSTCSS, set $\rho = 1/40$ and $\ell = \lceil \log(4d(m+1)/\rho) \rceil$. Let $K_0 = \lceil 16\ell \rceil$ and $L = C_W k\ell/\varepsilon$, where $C_W \ge 7200$ is an absolute constant. Each randomized preprocessing call uses failure probability $\eta = \rho/(4d+2)$. We independently assign the columns to $B_0 = \lceil m/K_0 \rceil$ buckets and use these buckets as the leaves of a balanced binary tree. All local calls and the final global call share the same removal set D and the same removal limit $M = \lceil (1+\varepsilon)m \rceil$.

The first step is to show that random partitioning gives every leaf a small valid input bound.

Lemma A.9. *With probability at least $1-\rho/2$, every leaf bucket U_b contains at most $4K_0$ optimal outliers. On this event, its total ridge leverage mass satisfies*

$$Z_\lambda(A_{U_b}) \le 2k + 4K_0.$$

Proof. Fix the optimal outlier set $N^\star$ before the random partition. For a fixed bucket, the number of indices from $N^\star$ is a sum of independent Bernoulli variables with expectation at most $m/B_0 \le K_0$. A Chernoff bound gives

$$\Pr[|N^\star \cap U_b| \ge 4K_0] \le (e/4)^{4K_0}.$$

Since $B_0 \le d$ and $K_0 = \lceil 16\ell \rceil$, a union bound over all leaf buckets is at most $\rho/2$.

On the resulting event, Lemma A.1 bounds the ridge leverage mass of the optimal inliers in U_b by $2k$. Every remaining optimal outlier contributes at most one by Lemma A.2. Since there are at most $4K_0$ such columns, the stated ridge-mass bound follows. □

The leaf bounds initiate the bottom-up hierarchy. Each internal node receives the surviving original columns returned by its two children. The following invariant shows that every RIDGEPEEL call is supplied with a valid ridge-mass upper bound.

Lemma A.10. *On the leaf-balance event and before the first failed preprocessing event, every reached node v receives an input set U_v satisfying the bound $Z_\lambda(A_{U_v}) \le H_v$ used by Algorithm 2. Then, every node that completes successfully returns a surviving set R_v with $Z_\lambda(A_{R_v}) \le L$.*

Proof. At a leaf, Lemma A.9 gives the valid input bound $H_v = 2k + 4K_0$. For the chosen constants, this bound is at most L, so the corresponding RIDGEPEEL call may return immediately. Its output therefore already has ridge leverage mass at most L.

Now consider an internal node with children v_1 and v_2. By induction, both child survivor sets have ridge leverage mass at most L. Their union is the input to the parent. Subadditivity from Lemma A.3 gives

$$Z_\lambda(A_{R_{v_1} \cup R_{v_2}}) \le Z_\lambda(A_{R_{v_1}}) + Z_\lambda(A_{R_{v_2}}) \le 2L.$$

Thus the choice $H_v = 2L$ used at every internal node is valid. If RIDGEPEEL returns after the score test, the deletion-stable estimates from Lemma A.5 imply $Z_\lambda(A_{R_v}) \le L$. If it returns through the condition $H_v \le L$, the same conclusion follows directly from the valid input bound. Induction proves the invariant throughout the hierarchy. □

The root returns a set R_0 with $Z_\lambda(A_{R_0}) \le L$. Algorithm 2 then invokes RIDGEPEEL once more on R_0, using input bound $H = L$ and the smaller target $K = 7200k/\varepsilon$. This final call reduces the ridge leverage mass to the level required by the subsequent column-selection step. The same set D records removals made throughout the hierarchy and during this final call.

It remains to show that all of these calls together remove only a small number of optimal inliers. The important point is that the analysis counts actual deletions globally; no separate inlier-removal allowance is assigned to individual nodes or tree levels.

Theorem A.11. *With probability at least $19/20 - \rho$, the hierarchical peeling stage and the final global call complete without reaching the removal limit M. The resulting set $R = [d] \setminus D$ satisfies $Z_\lambda(A_R) \le K$, and the total number of removed optimal inliers satisfies $|D \cap I^\star| \le \varepsilon m/10$.*

Proof. For the analysis, stop the process at the first failure of either the leaf-balance event or a randomized preprocessing event. This stopping rule is only analytical and need not be detected by the algorithm. Before the stopping time, Lemma A.10 provides a valid input mass bound for every reached call, while Lemma A.5 provides valid score estimates for every adaptively generated surviving set.

Consider the tth deletion before the stopping time. Let $\mathcal{F}_{t-1}$ denote the preceding history, R_t the current surviving set, and Λ_t the target of the current RIDGEPEEL call. A deletion occurs only when $Q_{R_t} > \Lambda_t$. The surviving optimal inliers contribute at most $18k$ estimated mass, and therefore

$$\Pr[J_t \in I^\star \mid \mathcal{F}_{t-1}] \le \frac{18k}{\Lambda_t}. \tag{7}$$

Every peeling target satisfies $\Lambda_t \ge K = 7200k/\varepsilon$, so the conditional probability above is at most $\varepsilon/400$. Let X be the number of optimal inliers removed before the stopped process terminates, with deletion indicators set to zero afterward. Since at most M deletions occur and $M \le 2m$,

$$\mathbb{E}[X] \le \frac{\varepsilon M}{400} \le \frac{\varepsilon m}{200}.$$

Then, applying Markov's inequality gives

$$\Pr\left[X > \varepsilon m/10\right] \le \frac{1}{20}.$$

On the event $X \le \varepsilon m/10$, the removal limit cannot be reached. Indeed, writing $\Delta = \lceil \varepsilon m \rceil$ gives $M = m + \Delta$. Removing M distinct columns would therefore remove at least

$$M - |N^\star| \ge M - m = \Delta$$

Algorithm 5 FASTCANDIDATECOVER

Input: surviving matrix A_R indexed by $R \subseteq [d]$ with $Z_\lambda(A_R) \le K$, ridge parameter $\lambda > 0$, accuracy $\varepsilon \in (0, 1/2]$, and failure probability $\eta \in (0, 1)$.

Output: a list C of sampled original indices and the corresponding reweighted candidate matrix $\widetilde{C}$, or FAIL.

if $R = \emptyset$ or $A_R = 0$ **then**
  return an empty index list and an empty matrix.
end if
Set $\alpha \leftarrow \varepsilon/128$.
Compute $u \leftarrow$ FASTAPPROXRIDGESCORES$(A_R, \lambda, K, \eta/3)$.
Set $\widehat{Z} \leftarrow \sum_{j \in R} u_j$.
if $\widehat{Z} \le 0$ **then**
  return FAIL.
end if
Set $s \leftarrow \left\lceil c_{\mathrm{cov}} \widehat{Z} \alpha^{-2} \log((K+1)/\eta) \right\rceil$, where $c_{\mathrm{cov}} > 0$ is a sufficiently small absolute constant.
if $s \ge |R|$ **then**
  return $C = R$ and $\widetilde{C} = A_R$.
end if
Set $p_j \leftarrow u_j / \widehat{Z}$ for every $j \in R$.
Independently sample $i_1, \dots, i_s$ according to $\{p_j\}_{j \in R}$.
Set $C \leftarrow (i_1, \dots, i_s)$ and construct $\widetilde{C} = \left[\frac{a_{i_1}}{\sqrt{s p_{i_1}}}, \dots, \frac{a_{i_s}}{\sqrt{s p_{i_s}}} \right]$.
return $(C, \widetilde{C})$.

optimal inliers, contradicting $X \le \varepsilon m/10 < \Delta$. Hence every reached RIDGEPEEL call terminates through its prescribed ridge-mass condition.

It remains to bound the probability that the stopped process is triggered by preprocessing. By Lemma A.9, the leaf-balance event fails with probability at most $\rho/2$. The hierarchy and final global call contain at most $2B_0 \le 2d$ preprocessing calls. Each reached call has conditional failure probability at most $\eta = \rho/(4d+2)$, so the probability that any such preprocessing call fails is less than $\rho/2$.

A union bound with the event $X > \varepsilon m/10$ therefore gives total failure probability at most $1/20 + \rho$. On the complementary event, all hierarchical calls reach their target, and the final global call reduces the ridge leverage mass to K. Hence $Z_\lambda(A_R) \le K$ and $|D \cap I^\star| \le \varepsilon m/10$. □

A.5 CANDIDATE COVER AND SPECTRAL COLUMN SELECTION

The preceding theorem supplies the input score bound for the column-selection steps. We collect here the two subroutines invoked by ROBUSTCSS and the sparsification subroutine they use. The following subroutines implement the final column-selection steps.

A.5.1 CANDIDATE SAMPLING AND RIDGE APPROXIMATION

After hierarchical peeling, the surviving matrix A_R satisfies $Z_\lambda(A_R) \le K$, but it may still contain a large number of columns. The goal of this step is to construct a much smaller candidate matrix that preserves the regularized covariance of A_R. This candidate matrix is used only as an intermediate representation for the subsequent top-tail sparsification; the final selected columns are still original columns of A_R. We construct the candidate matrix by sampling according to constant-factor ridge leverage-score estimates.

Algorithm 5 first uses FASTAPPROXRIDGESCORES, defined in Appendix A.2, to obtain upper estimates of the ridge leverage scores of A_R. It then samples columns independently in proportion to these estimates and rescales every sampled copy by its sampling probability. The number of samples is chosen so that the resulting regularized covariance approximates that of the entire surviving matrix.

The next lemma gives the property needed from this candidate sample. The regularization is important because the sampling probabilities are defined by ridge leverage scores. The resulting guarantee preserves the regularized covariance of the entire surviving matrix, including directions corresponding to columns that are not sampled.

Lemma A.12. *Let $B = A_R$ satisfy $Z_\lambda(B) \le K$. With probability at least $1-\eta$, Algorithm 5 returns a candidate matrix $\widetilde{C}$ satisfying*

$$(1-\alpha)(BB^\top + \lambda I_n) \preceq \widetilde{C}\widetilde{C}^\top + \lambda I_n \preceq (1+\alpha)(BB^\top + \lambda I_n), \tag{8}$$

where $\alpha = \varepsilon/128$. The number of sampled columns is $O\left(\frac{k}{\varepsilon^3}\log\frac{k}{\varepsilon\eta}\right)$.

Proof. Condition on the successful output of FASTAPPROXRIDGESCORES. Let u_j denote the resulting upper estimates and let $\widehat{Z} = \sum_j u_j$. By Lemma A.4, $\tau_j^\lambda(B) \le u_j$ for every column and $\widehat{Z} \le c_{\mathrm{lev}}K$. Define $\Gamma = (BB^\top + \lambda I_n)^{-1/2}$. For one sampled index $i \sim p$, where $p_j = u_j/\widehat{Z}$, the corresponding normalized rank-one contribution is $Y = \frac{1}{sp_i}\Gamma b_i b_i^\top \Gamma$. Its expectation is

$$\mathbb{E}[Y] = \frac{1}{s}\Gamma BB^\top \Gamma.$$

The ridge-score upper bound controls the spectral norm of every summand. Specifically, we have

$$\|Y\|_2 = \frac{\tau_i^\lambda(B)}{sp_i} \le \frac{u_i}{s(u_i/\widehat{Z})} = \frac{\widehat{Z}}{s}.$$

Moreover, $\mathrm{tr}(\Gamma BB^\top\Gamma) = Z_\lambda(B) \le K$. Therefore, standard matrix concentration applied to the sum of the s independent sampled contributions gives

$$\left\|\Gamma(BB^\top - \widetilde{C}\widetilde{C}^\top)\Gamma\right\|_2 \le \alpha$$

with probability at least $1 - 2\eta/3$, provided $s \ge c_{\mathrm{cov}}\widehat{Z}\alpha^{-2}\log((K+1)/\eta)$ for a sufficiently large absolute constant c_{cov}.

The normalized spectral bound implies Equation (8) after applying the congruence transformation with $(BB^\top + \lambda I_n)^{1/2}$. Combining this event with the failure probability $\eta/3$ of the ridge-score routine gives total success probability at least $1-\eta$.

Finally, $\widehat{Z} \le c_{\mathrm{lev}}K$ and $\alpha = \varepsilon/128$ give $s = O\left(\frac{K}{\varepsilon^2}\log\frac{K}{\eta}\right)$. Since $K = O(k/\varepsilon)$, the stated candidate-size bound follows. If $s \ge |R|$, the algorithm returns all columns of B. In this branch the covariance is preserved exactly and $|R| \le s$, so the same size bound continues to hold. □

The subsequent top-tail procedure also requires the candidate matrix itself to have controlled ridge leverage mass. This follows from the same sampling construction. On the successful score and candidate-cover events, the normalized trace of the sampled covariance satisfies

$$\mathrm{tr}\left(\Gamma\widetilde{C}\widetilde{C}^\top\Gamma\right) = \frac{1}{s}\sum_{t=1}^{s}\frac{\tau_{i_t}^\lambda(B)}{p_{i_t}} \le \widehat{Z} \le c_{\mathrm{lev}}K.$$

The lower inequality in Equation (8) gives $(\widetilde{C}\widetilde{C}^\top + \lambda I_n)^{-1} \preceq (1-\alpha)^{-1}(BB^\top + \lambda I_n)^{-1}$. Hence the total ridge leverage mass of the candidate matrix satisfies

$$Z_\lambda(\widetilde{C}) \le \frac{c_{\mathrm{lev}}K}{1-\alpha} \le 2c_{\mathrm{lev}}K. \tag{9}$$

If the algorithm returns all columns, then $\widetilde{C} = B$ and the stronger bound $Z_\lambda(\widetilde{C}) \le K$ holds directly. This completes the properties of the candidate matrix required by the next sparsification stage.

A.5.2 TOP-TAIL SELECTION VIA SPECTRAL SPARSIFICATION

The candidate cover still contains $\widetilde{O}(k/\varepsilon^3)$ sampled columns, while the desired selected-column bound is $O(k/\varepsilon^2)$. We further reduce this set using a BSS-style barrier sparsification argument. Our construction is a slight modification of the dual-set spectral sparsification framework of Boutsidis et al. Boutsidis et al. (2014), based on the BSS method of Batson et al. Batson et al. (2012). We use the standard lower-barrier construction and replace only the upper potential to obtain the spectral-tail guarantee required below. We therefore state the resulting sparsification guarantee and verify only the part that differs from the standard analysis.

Let $B = B_h + E$, where $B_h = U_h \Sigma_h V_h^\top$ is the rank-h truncated SVD of B. For each column position j, let $v_j = V_h^\top e_j$ denote its coordinate in the top right singular space, and let $g_j = Ee_j$ denote the corresponding tail column. These vectors satisfy $\sum_j v_j v_j^\top = I_h$.

Lemma A.13. *For every integer $r \geq 16h$, the modified BSS-style procedure selects at most r column positions with positive weights. Let Ω denote the corresponding sampling matrix and let D_{wt} denote the diagonal weight matrix. The selected top directions satisfy $\sigma_{\min}(V_h^\top \Omega D_{\mathrm{wt}}) \geq 1/2$, while the selected tail satisfies*

$$\|E\Omega D_{\mathrm{wt}}\|_2^2 \leq 4\|E\|_2^2 + \frac{4\|E\|_F^2}{r}.$$

If S is the set of distinct selected column indices, then for an absolute constant C_{tt},

$$\|(I - \Pi_{B_S})B\|_2^2 \leq C_{\mathrm{tt}} \left(\|E\|_2^2 + \frac{\|E\|_F^2}{r} \right). \tag{10}$$

Proof. We use the standard lower-barrier construction of Boutsidis et al. (2014). At every iteration, the corresponding quantities L_j satisfy $\sum_j L_j \geq 2/3$, and any update with $1/w \leq L_j$ preserves the lower spectral barrier. We only establish a compatible upper condition controlling the selected tail.

Let $W = EE^\top$, $\beta^2 = \|E\|_2^2$, and $F^2 = \|E\|_F^2$. If $E = 0$, the tail bound is immediate, and the standard lower-barrier updates alone give the claimed top-direction guarantee. We therefore assume $E \neq 0$.

Let $M_t = \sum_{s=1}^t w_s g_{j_s} g_{j_s}^\top$ and use the upper barrier $u_t = 4F^2 + 4t\beta^2$ with potential $\Psi_u(M) = \mathrm{tr}((uI - M)^{-1}W)$. Suppose $u_t I - M_t \succ 0$ and $\Psi_{u_t}(M_t) \leq 1/4$. Set $u' = u_t + 4\beta^2$, $N = (u'I - M_t)^{-1}$, and $\Delta_U = \Psi_{u_t}(M_t) - \Psi_{u'}(M_t)$. Define $U_j = \frac{g_j^\top NWNg_j}{\Delta_U} - g_j^\top N g_j$. By the Sherman–Morrison identity, any update satisfying $1/w > U_j$ preserves the upper invariant.

Let $P = (u_t I - M_t)^{-1}$. The resolvent identity gives $P - N \succeq 4\beta^2 N^2$, and hence

$$\Delta_U \geq 4\beta^2 \,\mathrm{tr}(N^2 W).$$

Using $\sum_j g_j g_j^\top = W \preceq \beta^2 I$, it holds that

$$\sum_j U_j \leq \frac{\beta^2 \,\mathrm{tr}(N^2 W)}{4\beta^2 \,\mathrm{tr}(N^2 W)} \leq \frac{1}{4}.$$

Since $\sum_j L_j \geq 2/3$, some index satisfies $L_j > U_j$. Choosing $w > 0$ with $U_j < 1/w \leq L_j$ therefore preserves both barriers.

After r iterations, the upper invariant gives

$$M_r \preceq (4F^2 + 4r\beta^2)I.$$

The standard lower-barrier guarantee gives

$$T_r \succeq (r - \sqrt{rh})I_h,$$

where $T_r = \sum_{s=1}^r w_s v_{j_s} v_{j_s}^\top$. Define $D_{\mathrm{wt}} = \mathrm{diag}\left(\sqrt{w_1/r}, \ldots, \sqrt{w_r/r}\right)$. Then, for $Y = V_h^\top \Omega D_{\mathrm{wt}}$, we have

$$YY^\top = \frac{T_r}{r} \succeq \left(1 - \sqrt{h/r}\right) I_h \succeq \frac{3}{4} I_h,$$

Algorithm 6 TOPTAILONCOVER

Input: reweighted candidate matrix $\widetilde{C}$, its corresponding list C of original indices, ridge parameter $\lambda > 0$, ridge-mass bound K, and accuracy $\varepsilon \in (0, 1/2]$.
Output: a set of original column indices $S \subseteq R$.

Set $r \leftarrow \lceil C_{\mathrm{tail}} K/\varepsilon \rceil$ and $\theta \leftarrow C_{\mathrm{tail}} \varepsilon \lambda$.
if C is empty **then**
  return $\emptyset$.
end if
if $|C| \le r$ **then**
  return the distinct original indices appearing in C.
end if
Compute a thin orthogonal factorization $\widetilde{C} = VT$ with $V^\top V = I$.
Compute the SVD of T and let h be the number of singular values satisfying $\sigma_i^2(T) > \theta$.
if $h = 0$ **then**
  return $\emptyset$.
end if
Let V_h contain the leading h right singular vectors of T, and let $E = T - T_h$.
Apply the modified BSS-style sparsification from Lemma A.13 to (V_h, E, r).
Map the selected positions through C, and let S be the set of distinct original indices obtained.
return S.

since $r \ge 16h$. Thus, we get $\sigma_{\min}(Y) \ge 1/2$. The same normalization gives

$$\|E\Omega D_{\mathrm{wt}}\|_2^2 \le 4\beta^2 + \frac{4F^2}{r}.$$

Finally, let $C = B\Omega D_{\mathrm{wt}}$. Since $C = U_h \Sigma_h Y + E\Omega D_{\mathrm{wt}}$ and Y has full row rank,

$$(I - \Pi_C)B_h = -(I - \Pi_C)E\Omega D_{\mathrm{wt}} Y^\dagger V_h^\top.$$

Therefore, we have

$$\|(I - \Pi_C)B_h\|_2 \le 2\|E\Omega D_{\mathrm{wt}}\|_2.$$

Combining this with $B = B_h + E$ and the tail bound proves Equation (10) for an absolute constant C_{tt}. Reweighting and repeated selections do not change the selected column span, so $\Pi_C = \Pi_{B_S}$. □

Applying sparsification to the candidate cover. We now apply Lemma A.13 to the reweighted candidate matrix returned by FASTCANDIDATECOVER. The candidate matrix may contain rescaled and repeated copies of original columns. We therefore perform the sparsification in a low-dimensional coordinate representation and map the selected positions back to their original indices.

The factorization $\widetilde{C} = VT$ is used only to reduce the dimension of the sparsification problem. Since V has orthonormal columns, projection errors for selected spans are preserved between T and $\widetilde{C}$. Mapping the selected positions through the list C recovers original columns of A_R. Repeated copies and positive rescalings do not affect the resulting column span.

We next verify that the candidate matrix satisfies the conditions of Lemma A.13. Let $K_C = 2c_{\mathrm{lev}}K$. By Equation (9), $Z_\lambda(T) = Z_\lambda(\widetilde{C}) \le K_C$. Since every retained singular direction satisfies $\sigma_i^2(T) > \theta$, its contribution to the ridge leverage mass is at least $\theta/(\lambda + \theta)$. Therefore, the number of retained directions satisfies $h \le \frac{\lambda+\theta}{\theta} K_C = O(K/\varepsilon)$. Choosing C_{tail} sufficiently large guarantees $r \ge 16h$.

The remaining singular values are at most θ, so the spectral tail satisfies $\|E\|_2^2 \le \theta$. The same ridge-mass bound also controls its Frobenius norm

$$\|E\|_F^2 \le (\lambda + \theta)K_C.$$

Lemma A.14. *Suppose $B = A_R$ satisfies $Z_\lambda(B) \le K$ and the candidate-cover event in Lemma A.12 holds. Then Algorithm 6 returns a set $S \subseteq R$ with $|S| = O(K/\varepsilon) = O(k/\varepsilon^2)$. Moreover, the selected span satisfies*

$$\|(I - \Pi_{A_S})B\|_2^2 \le \frac{\varepsilon\lambda}{16}.$$

Proof. Lemma A.13 applied to the coordinate matrix T gives

$$\|(I - \Pi_{A_S})\widetilde{C}\|_2^2 \le C_{\mathrm{tt}} \left(\|E\|_2^2 + \frac{\|E\|_F^2}{r} \right) \le C_{\mathrm{tt}} \left(\theta + \frac{(\lambda+\theta)K_C}{r} \right).$$

Since $\theta = c_{\mathrm{tail}}\varepsilon\lambda$ and $r = C_{\mathrm{tail}}K/\varepsilon$ up to rounding, choosing c_{tail} sufficiently small and C_{tail} sufficiently large gives

$$\|(I - \Pi_{A_S})\widetilde{C}\|_2^2 \le \frac{\varepsilon\lambda}{32}.$$

If $h = 0$, the empty set already gives cover error at most θ. If $|C| \le r$, returning all distinct candidate indices gives zero cover error. Thus the same bound holds in every branch of Algorithm 6.

It remains to transfer the spectral guarantee from the candidate matrix back to B. Let $Q = I - \Pi_{A_S}$. The lower ridge approximation from Lemma A.12 gives

$$(1-\alpha)\left(QBB^\top Q + \lambda Q\right) \preceq Q\widetilde{C}\widetilde{C}^\top Q + \lambda Q.$$

Taking the largest Rayleigh quotient over the range of Q gives

$$\|QB\|_2^2 \le \frac{\|Q\widetilde{C}\|_2^2 + \alpha\lambda}{1-\alpha}.$$

Using $\alpha = \varepsilon/128$ and the preceding candidate-cover bound, we obtain

$$\|QB\|_2^2 \le \frac{\varepsilon\lambda/32 + \varepsilon\lambda/128}{1-\varepsilon/128} \le \frac{\varepsilon\lambda}{16}.$$

Finally, the sparsifier selects at most $r = O(K/\varepsilon)$ positions, so the number of distinct original indices is also $O(K/\varepsilon) = O(k/\varepsilon^2)$. □

A.6 Proof of Theorem 3.5

We now complete the proof of the bicriteria guarantee. The preceding sections establish the two properties needed from peeling and column selection. With constant probability, hierarchical peeling returns $R = [d] \setminus D$ satisfying $Z_\lambda(A_R) \le K$ and $|D \cap I^\star| \le \varepsilon m/10$. Conditioned on this event, Lemma A.14 returns $S \subseteq R$ with $|S| = O(k/\varepsilon^2)$ and $\|(I_n - \Pi_{A_S})A_R\|_2^2 \le \varepsilon\lambda/16$.

Proof of Theorem 3.5. Let $J = R \cap I^\star$ denote the surviving optimal inliers and set $Q = I_n - \Pi_{A_S}$. Since $J \subseteq R$, the spectral guarantee on A_R also gives $\|QA_J\|_2^2 \le \varepsilon\lambda/16$. Moreover, since $J \subseteq I^\star$ and $\dim(U^\star) \le k$, the subspace $U^\star$ is a feasible rank-k subspace for A_J. Hence, we have

$$\|A_J - (A_J)_k\|_F^2 \le \|(I_n - P^\star)A_J\|_F^2 \le \|(I_n - P^\star)A_{I^\star}\|_F^2 = f^\star.$$

To convert this spectral bound into Frobenius error, we observe that $\sigma_i(QA_J) \le \sigma_i(A_J)$ for every i. Therefore, we get

$$\|QA_J\|_F^2 = \sum_i \sigma_i(QA_J)^2 \le \sum_i \min\left\{ \sigma_i(A_J)^2, \frac{\varepsilon\lambda}{16} \right\} \le \|A_J - (A_J)_k\|_F^2 + \frac{k\varepsilon\lambda}{16} \le (1+\varepsilon/8)f^\star.$$

We next transfer this guarantee to the original matrix. Let $H = D \cap I^\star$ contain the optimal inliers removed during peeling, and consider the comparison deletion set $F = (N^\star \cup H) \setminus S$. The peeling guarantee gives $|F| \le m + \varepsilon m/10 \le M$, and F is disjoint from S. After deleting F, every remaining column outside S belongs to J, so the resulting reconstruction error is at most $\|QA_J\|_F^2$.

Algorithm 2 instead removes the $\min\{M, d - |S|\}$ largest residuals among the indices outside S. For the fixed span $\mathrm{span}(A_S)$, this choice maximizes the total removed residual over all eligible deletion sets of size at most M. Its remaining error is therefore no larger than the error obtained by deleting F. Consequently, the returned set N satisfies

$$\|(I_n - \Pi_{A_S})A_{[d]\setminus N}\|_F^2 \le (1+\varepsilon/8)f^\star \le (1+\varepsilon)f^\star.$$

By construction, $S \cap N = \emptyset$ and $|N| \le M = (1+\varepsilon)m$.

For $m \geq 1$, Theorem A.11 gives the peeling event with probability at least $19/20 - \rho$. Conditional on this event, the candidate-cover construction fails with probability at most ρ, while the top-tail selection is deterministic after the cover is fixed. Taking $\rho = 1/40$, all required events hold with probability at least $9/10$. When $m = 0$, peeling is skipped and $Z_\lambda(A) \leq 2k \leq K$, so the same column-selection and residual trimming argument applies.

We finally bound the running time of one execution. Let $H_* = \widetilde{O}(k/\varepsilon)$ be the largest ridge-mass bound used during hierarchical peeling. By the preprocessing bound above and Lemma A.8, summing over the hierarchy gives $\widetilde{O}\left(\frac{ndk^2}{\varepsilon^2} + d\left(\frac{k}{\varepsilon}\right)^3\right)$ time, including all deletion updates, since each column is removed at most once. For column selection, the candidate cover has size $s = \widetilde{O}(k/\varepsilon^3)$ and the final selected set has size $r = O(k/\varepsilon^2)$. The preceding candidate-construction, sparsification, and residual-trimming bounds give $\widetilde{O}\left(\frac{ndk^2}{\varepsilon^2} + \frac{nk^2}{\varepsilon^6} + \frac{k^4}{\varepsilon^{11}}\right)$. Combining the two stages yields $\widetilde{O}\left(\frac{ndk^2}{\varepsilon^2} + d\left(\frac{k}{\varepsilon}\right)^3 + \frac{nk^2}{\varepsilon^6} + \frac{k^4}{\varepsilon^{11}}\right)$, which matches Theorem 3.5. □

B Proofs for the Exact-k Approximation

This section provides the proof and implementation details for Algorithm 3 and Theorem 4.4. We first analyze the exact ridge-score peeling step and reduce the surviving matrix to $O(k/\varepsilon)$ dimensions. We then construct a strong coreset for the truncated-residual objective and solve the compressed problem exactly. Finally, we show that thresholding controls both the number of removed columns and the rank-k error of the retained matrix, after which a standard exact-k CSS routine completes the construction.

Throughout this section, $m \geq 1$, $0 < \varepsilon \leq 1/2$, and $f^\star \leq \widehat{f} \leq 2f^\star$ with $f^\star > 0$. We use $\Delta = \lceil \varepsilon m \rceil$, $M = m + \Delta$, $\lambda = \widehat{f}/k$, and $K = C_0 k/\varepsilon$, where C_0 is a sufficiently large absolute constant.

B.1 Exact-Score Peeling and Low-Dimensional Reduction

The first stage of Algorithm 3 removes columns until the total ridge leverage mass is at most $K = O(k/\varepsilon)$. Since this stage computes the ridge leverage scores of the current surviving matrix exactly, the optimal-inlier mass bound from Lemma A.1 directly controls the probability of removing an optimal inlier.

Lemma B.1. *With probability at least $7/8$, the peeling stage removes at most $\Delta/4$ optimal inliers. On the same event, the removal limit M is not reached and the returned set R satisfies $Z_\lambda(A_R) \leq K$.*

Proof. Let $\mathcal{F}_{t-1}$ denote the history before the tth deletion, and let R_t be the current surviving set. A deletion occurs only when $Z_\lambda(A_{R_t}) > K$. By Lemma A.1, the surviving optimal inliers have total ridge leverage mass at most $2k$. Since deletion is proportional to the exact ridge leverage scores,

$$\Pr[J_t \in I^\star \mid \mathcal{F}_{t-1}] \leq \frac{2k}{Z_\lambda(A_{R_t})} \leq \frac{2k}{K} = \frac{2\varepsilon}{C_0}.$$

Let $X = |D \cap I^\star|$ be the number of optimal inliers removed. There are at most M deletions, so the conditional bound above gives

$$\mathbb{E}[X] \leq \frac{2\varepsilon M}{C_0} \leq \frac{4\Delta}{C_0},$$

where $M \leq 2m$ and $\Delta \geq \varepsilon m$. Therefore, Markov's inequality yields

$$\Pr[X > \Delta/4] \leq \frac{16}{C_0} \leq \frac{1}{8},$$

where the last inequality uses $C_0 \geq 128$.

It remains to show that on the event $X \leq \Delta/4$, the removal limit cannot be reached. Since $M = \lceil (1+\varepsilon)m \rceil = m + \Delta$, removing M distinct columns would remove at least

$$M - |N^\star| \geq M - m = \Delta$$

optimal inliers, contradicting $X \leq \Delta/4$. Hence the peeling loop terminates before reaching the removal limit and must satisfy $Z_\lambda(A_R) \leq K$. □

The same ridge-mass bound reduces the dimension of the subsequent subspace search. Let $V \in \mathbb{R}^{n\times r}$ contain the left singular vectors of A_R whose squared singular values exceed λ.

Lemma B.2. *The retained subspace has dimension $r \le 2K = O(k/\varepsilon)$. Its projection error satisfies*

$$\|(I_n - VV^\top)A_R\|_F^2 \le 2\lambda K = O(f^\star/\varepsilon).$$

Proof. Let $\sigma_i = \sigma_i(A_R)$. Every retained singular direction satisfies $\sigma_i^2 > \lambda$ and therefore contributes more than $1/2$ to $Z_\lambda(A_R) = \sum_i \sigma_i^2/(\sigma_i^2+\lambda)$. Thus, $r < 2Z_\lambda(A_R) \le 2K$. For every omitted singular direction, $\sigma_i^2 \le \lambda$, which implies

$$\sigma_i^2 \le 2\lambda \frac{\sigma_i^2}{\sigma_i^2+\lambda}.$$

Summing over the omitted directions gives

$$\|(I_n - VV^\top)A_R\|_F^2 \le 2\lambda Z_\lambda(A_R) \le 2\lambda K.$$

Since $\lambda = \widehat{f}/k$, $K = C_0 k/\varepsilon$, and $\widehat{f} \le 2f^\star$, the final quantity is $O(f^\star/\varepsilon)$. □

B.2 A Strong Coreset for truncated Residuals

Let $z_j = V^\top a_j$ for every $j \in R$, and set $\theta = 64\widehat{f}/\Delta$. For every subspace $W \subseteq \mathbb{R}^r$ with $\dim(W) \le k$, define

$$\Phi_\theta(W) = \sum_{j\in R} \min\left\{\|(I_r - \Pi_W)z_j\|_2^2, \theta\right\}.$$

The goal of this section is to construct a weighted sample that preserves $\Phi_\theta(W)$ simultaneously for all candidate subspaces. Since $r = O(k/\varepsilon)$, we allow an exponential dependence on r^2 while keeping the dependence on the original number of columns polylogarithmic.

For a rank-at-most-k orthogonal projector X, the squared residual of z_j is $\|z_j\|_2^2 - \langle z_j z_j^\top, X\rangle$. Let $p = r(r+1)/2 + 1$. Using an isometric symmetric vectorization svec, define

$$u_j = \begin{bmatrix} -\mathrm{svec}(z_j z_j^\top) \\ \|z_j\|_2^2 \end{bmatrix} \in \mathbb{R}^p.$$

For a projector X, define

$$y_X = \begin{bmatrix} \mathrm{svec}(X) \\ 1 \end{bmatrix}.$$

The lifted inner product then satisfies $u_j^\top y_X = \|(I_r - X)z_j\|_2^2$.

For the sensitivity analysis, we consider the enlarged query family $g_j(y) = \min\{|u_j^\top y|, \theta\}$ over $y \in \mathbb{R}^p$. Every projector query belongs to this family, so uniform preservation of these functions implies uniform preservation of the truncated-residual objective.

Lemma B.3. *The nonzero lifted vectors can be partitioned into layers $B_1, B_2, \ldots$, each containing at most p vectors, such that every vector assigned to layer B_t has sensitivity at most $2^p/t$. Consequently, the total sensitivity satisfies $\mathcal{S} = O\left(2^p p \log(d+2)\right)$.*

Proof. Consider a current collection Q of nonzero lifted vectors. Construct a basis $B = \{b_1, \ldots, b_h\}$ by pivoted Gram–Schmidt, always choosing the vector with maximum distance from the span of the previously selected vectors. Here $h \le p$. Let $q_1, \ldots, q_h$ be the resulting orthonormal directions, and let δ_t be the residual norm of b_t when it is selected.

The maximal choice of each pivot implies $|q_t^\top u| \le \delta_t$ for every $u \in Q$. Expressing u in the selected basis and normalizing the corresponding triangular system by δ_t shows by backward substitution that $u = \sum_{\ell=1}^h c_\ell b_\ell$ with $\sum_{\ell=1}^h |c_\ell| \le 2^h - 1 \le 2^p$. Therefore, every query y satisfies $g_u(y) \le 2^p \max_{b\in B} g_b(y)$.

Algorithm 7 TRUNCATEDRESIDUALCORESET

Input: projected columns $(z_j)_{j\in R}$, cap $\theta > 0$, accuracy $\xi \in (0,1)$, and failure probability $\delta_c \in (0,1)$.
Output: a weighted coreset for the truncated-residual objective.

```
Form the lifted vectors u_j and omit zero vectors.
Initialize the remaining collection Q and set t ← 1.
if Q = ∅ then
    return an empty weighted coreset.
end if
while Q is nonempty do
    Construct a pivoted Gram–Schmidt basis B_t of Q.
    Assign s_j ← 2^p/t to each index in B_t.
    Remove B_t from Q and set t ← t + 1.
end while
Set S ← Σ_j s_j.
Choose the sample size s_C according to Lemma B.4.
Sample s_C indices independently with probability p_j = s_j/S.
Give each sampled copy of z_j weight 1/(s_C p_j) and combine repeated copies.
return the resulting weighted projected columns.
```

We remove this basis and repeat the construction on the remaining vectors, producing layers $B_1, B_2, \ldots$. If $u_j \in B_t$, then u_j was present during each of the first t basis constructions. Applying the preceding bound to all these layers gives

$$t\, g_j(y) \le 2^p \sum_{\ell=1}^{t} \sum_{b\in B_\ell} g_b(y) \le 2^p \sum_i g_i(y).$$

Thus, the sensitivity of j is at most $2^p/t$. Since every layer contains at most p vectors, we have

$$\mathcal{S} \le 2^p p \sum_{t=1}^{d} \frac{1}{t} = O\left(2^p p \log(d+2)\right).$$

□

The preceding construction directly provides sensitivity upper bounds. A vector in layer B_t is assigned $s_j = 2^p/t$. We sample according to these values and preserve the original projected vectors z_j with weights. The vectors themselves are not rescaled because rescaling would change the cap θ. If all lifted vectors are zero, then all projected columns are zero and $\Phi_\theta(W) = 0$ for every feasible subspace W. In this case, the empty weighted coreset preserves the objective exactly. Otherwise, the total sampling weight S is positive, and the following sampling argument applies.

Lemma B.4. *For every $\xi, \delta_c \in (0,1)$, Algorithm 7 constructs a weighted coreset of size*

$$2^{O(p)}\, \mathrm{poly}\left(p, \frac{1}{\xi}\right) \log(d+2) \left(p \log p + \log\log(d+2) + \log\frac{1}{\delta_c}\right)$$

such that, with probability at least $1-\delta_c$,

$$(1-\xi)\Phi_\theta(W) \le \widetilde{\Phi}_\theta(W) \le (1+\xi)\Phi_\theta(W)$$

simultaneously for every subspace $W \subseteq \mathbb{R}^r$ with $\dim(W) \le k$. Since $p = O(k^2/\varepsilon^2)$, the coreset size is

$$2^{O(k^2/\varepsilon^2)}\, \mathrm{poly}(k, 1/\varepsilon, 1/\xi)\, \mathrm{polylog}(d+2).$$

Proof. Lemma B.3 gives $\mathcal{S} = O(2^p p \log(d+2))$. It remains to bound the combinatorial dimension of the enlarged query family $g_y(u) = \min\{|u^\top y|, \theta\}$.

For any threshold $a < \theta$, the condition $g_y(u) \ge a$ is equivalent to

$$(u^\top y)^2 - a^2 \ge 0.$$

This is a degree-two polynomial inequality in the p query variables, so standard sign-pattern bounds give combinatorial dimension $O(p \log p)$. Applying the standard sensitivity-sampling theorem with the sensitivity bounds above gives the stated sample size and uniform approximation. Since the projector queries form a subset of the enlarged family, the guarantee applies to every subspace of dimension at most k. □

B.3 Exact Optimization of the truncated Coreset

The strong coreset reduces the number of columns involved in the subspace search, while the truncated-residual objective remains nonconvex. We show that the coreset objective can be solved exactly by enumerating realizable residual-threshold patterns. The key observation is that once the set of columns below the threshold is fixed, the corresponding candidate is given by an ordinary weighted PCA solution.

Suppose the coreset consists of weighted projected columns $(z_i, w_i)_{i=1}^{s_C}$. For a rank-at-most-k orthogonal projector X, let $r_i(X) = z_i^\top (I_r - X) z_i$.

Lemma B.5. *There exists an optimal projector for the truncated coreset objective that is a weighted rank-k PCA solution for some subset of the coreset indices.*

Proof. Let $X^\star$ minimize the truncated-residual objective and define $G^\star = \{i : r_i(X^\star) < \theta\}$, the set of columns whose residuals lie below the threshold. Let $\overline{X}$ minimize the ordinary weighted PCA loss $\sum_{i \in G^\star} w_i r_i(X)$ over all rank-at-most-k projectors. The truncated cost of $\overline{X}$ satisfies

$$\begin{aligned}\sum_i w_i \min\{r_i(\overline{X}), \theta\} &\le \sum_{i \in G^\star} w_i r_i(\overline{X}) + \theta \sum_{i \notin G^\star} w_i \\ &\le \sum_{i \in G^\star} w_i r_i(X^\star) + \theta \sum_{i \notin G^\star} w_i \\ &= \sum_i w_i \min\{r_i(X^\star), \theta\}.\end{aligned}$$

Thus $\overline{X}$ is also globally optimal. □

Each threshold equation $r_i(X) = \theta$ is affine in the $r(r+1)/2$ independent entries of the symmetric matrix X. Hence the coreset induces an arrangement of s_C hyperplanes in $O(r^2)$ dimensions with at most $s_C^{O(r^2)}$ faces. We enumerate these faces, construct a weighted PCA candidate from the columns whose residuals lie below the threshold in each face, and evaluate every candidate using the full truncated-residual coreset objective.

The PCA candidate obtained from a face is not required to remain in that face. Exactness only requires that the below-threshold set induced by an optimal projector appears among the enumerated patterns. Lemma B.5 then guarantees that the weighted PCA candidate associated with this set is also globally optimal.

Let $q = r(r+1)/2$. Since $r = O(k/\varepsilon)$, we have $q = O(k^2/\varepsilon^2)$. The coreset size from Lemma B.4 gives

$$s_C^{O(q)} = 2^{O(k^4/\varepsilon^4)} (\mathrm{polylog}(d+2))^{O(k^2/\varepsilon^2)} \, \mathrm{poly}(k, 1/\varepsilon).$$

For every integer $t \ge 1$ and $d \ge 2$, $(\log d)^t \le (t/e)^t d$. The polylogarithmic factor can therefore be absorbed into the parameter-dependent term and a fixed-degree polynomial in d. Consequently, the exact coreset search runs in $2^{O(k^4/\varepsilon^4)} \, \mathrm{poly}(d)$ time.

B.4 Thresholding and Proof of Theorem 4.4

We now return from the compressed optimization problem to the original columns. Let $\widehat{W}$ be the exact minimizer returned by Algorithm 8, using $\xi = \varepsilon/256$. Define B as the set of surviving columns whose projected squared residual reaches the cap,

$$B = \left\{ j \in R : \|(I_r - \Pi_{\widehat{W}}) z_j\|_2^2 \ge \theta \right\},$$

Algorithm 8 SOLVETRUNCATEDCORESET

Input: weighted projected columns $(z_i, w_i)_{i=1}^{s_C}$, cap $\theta > 0$, and target dimension k.

Output: a subspace $\widehat{W}$ of dimension at most k minimizing the truncated coreset objective.

if $s_C = 0$ **then**
  return the zero subspace.
end if
Form the threshold equations $z_i^\top (I_r - X) z_i = \theta$ in the symmetric entries of X.
Enumerate all faces of the induced hyperplane arrangement.
for each face **do**
  Let G contain the indices whose residual expression is strictly below θ on this face.
  Compute the weighted rank-at-most-k PCA projector X_G for the columns indexed by G.
  Evaluate $\sum_i w_i \min\{z_i^\top (I_r - X_G) z_i, \theta\}$.
end for
return the range of a minimum-cost candidate.

and let $T = R \setminus B$. The truncated objective will simultaneously control the number of columns in B and the approximation error of A_T.

Let $q_R = |R \cap N^\star|$ denote the number of optimal outliers that remain after peeling. Define $W_0 \coloneqq V^\top U^\star \subseteq \mathbb{R}^r$. Since $\dim(U^\star) \le k$, we have $\dim(\mathrm{W}_0) \le k$. Moreover, for every $j \in R \cap I^\star$, it holds that

$$\|(I_r - \Pi_{W_0}) z_j\|_2 \le \|V^\top (a_j - P^\star a_j)\|_2 \le \|(I_n - P^\star) a_j\|_2.$$

Therefore, we have

$$\mathrm{OPT}_{\mathrm{cap}} \coloneqq \min_{\dim(W_0) \le k} \Phi_\theta(W) \le \Phi_\theta(W_0) \le \sum_{j \in R \cap I^\star} \|(I_n - P^\star) a_j\|_2^2 + q_R \theta \le f^\star + q_R \theta.$$

Optimality of $\widehat{W}$ on the coreset and the strong coreset guarantee give

$$\Phi_\theta(\widehat{W}) \le \beta \, \mathrm{OPT}_{\mathrm{cap}},$$

where $\beta = (1+\xi)/(1-\xi)$. Since $\xi = \varepsilon/256$ and $\varepsilon \le 1/2$, we have $\beta - 1 \le \varepsilon/64$ and $\beta \le 2$.

Lemma B.6. *Conditioned on successful peeling and coreset construction, the thresholded set satisfies $|D \cup B| \le M$. Moreover, the retained matrix satisfies*

$$\|A_T - (A_T)_k\|_F^2 = O(f^\star/\varepsilon).$$

Proof. Every index in B contributes exactly θ to the truncated objective. Therefore, it holds that $|B| \le \frac{\Phi_\theta(\widehat{W})}{\theta} \le \beta q_R + \beta \frac{f^\star}{\theta}$. The choice $\theta = 64\widehat{f}/\Delta$ gives $(\beta - 1) q_R \le \Delta/64$, $\beta f^\star/\theta \le \Delta/32$, and $|B| < q_R + \frac{\Delta}{16}$.

Let $h = |D \cap I^\star|$. On the good peeling event, $h \le \Delta/4$. Since $D = [d] \setminus R$, the number of optimal outliers removed during peeling is $|N^\star| - q_R$. Thus, we get $|D| = h + |N^\star| - q_R$. Combining this identity with the bound on $|B|$ gives

$$|D \cup B| = |D| + |B| < h + |N^\star| + \frac{\Delta}{16} \le m + \frac{5\Delta}{16} < M,$$

where D and B are disjoint because $B \subseteq R$.

It remains to bound the rank-k error of the retained matrix. By definition of T, every retained column has squared residual below θ. Hence, we have

$$\sum_{j \in T} \|(I_r - \Pi_{\widehat{W}}) z_j\|_2^2 \le \Phi_\theta(\widehat{W}).$$

Using the comparison bound for the truncated-residual objective and $\beta \le 2$, we obtain

$$\Phi_\theta(\widehat{W}) \le 2 f^\star + 2 q_R \theta = O(f^\star/\varepsilon),$$

where $q_R \le m$, $\Delta \ge \varepsilon m$, and $\widehat{f} \le 2f^\star$.

Define the corresponding subspace in the original space by $\widetilde{W} = V\widehat{W}$. For each retained column, the squared residual decomposes into the projection loss outside $\mathrm{span}(V)$ and the residual inside the projected space. Therefore, it follows that

$$\sum_{j\in T} \|(I_n - \Pi_{\widetilde{W}})a_j\|_2^2 \le \|(I_n - VV^\top)A_R\|_F^2 + \Phi_\theta(\widehat{W}) = O(f^\star/\varepsilon),$$

where Lemma B.2 controls the first term. Since $\dim(\widetilde{W}) \le k$, the best rank-$k$ approximation to A_T has no larger error, proving the second claim. □

Lemma B.7. *Let $B \in \mathbb{R}^{n\times t}$ and $1 \le k \le t$. The routine* VOLUMECSS(B, k) *returns a set $S \subseteq [t]$ with $|S| = k$ such that, with probability at least $1/2$, $\|(I_n - \Pi_{B_S})B\|_F^2 \le 2(k+1)\|B - B_k\|_F^2$. If $\mathrm{rank}(B) \le k$, the routine returns a set with zero reconstruction error deterministically. The routine can be implemented using $O(ktn^\omega \log(n+1))$ arithmetic operations, where ω denotes the matrix multiplication exponent.*

Proof. If $\mathrm{rank}(B) \le k$, we select a column basis of B and add arbitrary distinct column indices until $|S| = k$. Then, we have $\mathrm{span}(B_S) = \mathrm{span}(B)$, so the reconstruction error is zero.

Otherwise, we apply the exact volume-sampling algorithm of Deshpande & Rademacher (2010, Theorem 7) to $B^\top$. The selected rows of $B^\top$ correspond to a set S of exactly k columns of B. By the volume-sampling error bound stated in Deshpande & Rademacher (2010, Theorem 2),

$$\mathbb{E}_S\left[\|(I_n - \Pi_{B_S})B\|_F^2\right] \le (k+1)\|B - B_k\|_F^2.$$

Since $\mathrm{rank}(B) > k$, the quantity $\|B - B_k\|_F^2$ is positive. Using Markov's inequality gives

$$\Pr\left[\|(I_n - \Pi_{B_S})B\|_F^2 \le 2(k+1)\|B - B_k\|_F^2\right] \ge \frac{1}{2}.$$

The rank computation and column-basis extraction require $O(nt\min\{n,t\})$ arithmetic operations. For the other branch, Theorem 7 of Deshpande & Rademacher (2010), applied to the $t \times n$ matrix $B^\top$, gives the sampling cost $O(ktn^\omega \log(n+1))$. This also bounds the rank-computation and basis-selection cost, proving the claimed running time. □

Proof of Theorem 4.4. Let $\Delta = \lceil \varepsilon m \rceil$ and $M = m + \Delta = (1+\varepsilon)m$, where M is the intermediate peeling removal limit. Let $\mathcal{E}$ be the joint event that peeling and coreset construction succeed. By their guarantees, $\mathcal{E}$ has probability bounded below by a positive absolute constant. Condition on $\mathcal{E}$. Lemma B.6 gives $\|A_T - (A_T)_k\|_F^2 = O(f^\star/\varepsilon)$. Moreover, the removal-count estimate in its proof gives $|D \cup B| < m + \frac{5\Delta}{16} < M$. Since $|D \cup B|$ is an integer and $M = \lceil (1+\varepsilon)m \rceil$, it follows that

$$|D \cup B| \le (1+\varepsilon)m.$$

We first consider the case $|T| \ge k$. Using fresh randomness, apply VOLUMECSS(A_T, k) from Lemma B.7. For every fixed realization of the preceding stages, the routine returns, with conditional probability at least $1/2$, a set $S \subseteq T$ with $|S| = k$ such that

$$\|(I_n - \Pi_{A_S})A_T\|_F^2 \le 2(k+1)\|A_T - (A_T)_k\|_F^2 = O(k/\varepsilon)f^\star.$$

If $\mathrm{rank}(A_T) \le k$, the same bound holds deterministically with zero reconstruction error.

If $|T| < k$, Algorithm 3 selects all columns in T and adds $k - |T|$ distinct original column indices from $[d] \setminus T$. Such indices exist because $k \le d$. The resulting set S has size k and contains T, so the residual error on A_T is zero.

We next transfer the guarantee to the original matrix. In both cases, define the comparison deletion set

$$F = (D \cup B) \setminus S.$$

Then, $F \cap S = \emptyset$ and $|F| \le |D \cup B| \le (1+\varepsilon)m$. Since $T = [d] \setminus (D \cup B)$, deleting F leaves precisely the columns indexed by $T \cup S$. Every column indexed by S has zero residual with respect to $\mathrm{span}(A_S)$. Hence, we get

$$\|(I_n - \Pi_{A_S})A_{[d]\setminus F}\|_F^2 = \|(I_n - \Pi_{A_S})A_T\|_F^2.$$

The final trimming step chooses as many largest-residual columns outside S as possible subject to $|N| \leq (1+\varepsilon)m$. For the fixed selected span, this choice maximizes the total removed residual over all eligible deletion sets satisfying this cardinality constraint. Since F is such a feasible set, it holds that

$$\begin{aligned}\|(I_n - \Pi_{A_S})A_{[d]\setminus N}\|_F^2 &\leq \|(I_n - \Pi_{A_S})A_{[d]\setminus F}\|_F^2 \\ &= \|(I_n - \Pi_{A_S})A_T\|_F^2 \\ &\leq O(k/\varepsilon)f^\star.\end{aligned}$$

Thus, we obtain $|S| = k$, $S \cap N = \emptyset$, and $|N| \leq (1+\varepsilon)m$.

We bound the running time of one execution given $f^\star \leq \widehat{f} \leq 2f^\star$ under the exact arithmetic model, where an eigendecomposition of an $s \times s$ symmetric matrix costs $O(s^3)$ operations. Specifically, the exact-score peeling stage performs at most $O(m)$ iterations. Each iteration computes the ridge leverage scores in $O(n^3 + n^2 d)$ time, giving $O(m(n^3 + n^2 d))$ time in total. The subsequent spectral projection is absorbed by this bound since $m \geq 1$. Then, the coreset construction and exact search require $2^{O(k^4/\varepsilon^4)}d^2$ time by the preceding bounds. If $|T| \geq k$, the standard VOLUMECSS routine requires $O(kdn^\omega \log(n+1))$ time by Lemma B.7; otherwise, the padding step is lower order. The final residual computation and trimming are also absorbed by the preceding terms. Hence, one execution runs in $O\left(2^{O(k^4/\varepsilon^4)}d^2 + m(n^3 + n^2 d) + kdn^\omega \log n\right)$ time. This is fixed-parameter tractable in k and $1/\varepsilon$. The preceding success events occur with constant probability, and conditioned on them, VOLUMECSS succeeds with probability at least $1/2$ when $|T| \geq k$ and deterministically when $|T| < k$. Therefore, one execution succeeds with constant probability. □