5. Supplementary Material

5.1. Proof of Theorem 1.5

In this subsection we give the detailed proof of our hardness result, Theorem 1.5, which we restate here for convenience. Recall that ROBUST SUBSPACE RECOVERY is the special case of PCA WITH OUTLIERS where the task is to represent $\mathbf{A}$ exactly as the sum of a low-rank matrix and an outlier matrix.

Theorem 1.5. *There is no algorithm solving* ROBUST SUBSPACE RECOVERY *in time $f(d) \cdot n^{o(d)}$ for any computable function f of d, unless ETH fails.*

Proof. We show a reduction from CLIQUE. First, we need a set of points satisfying a certain generality condition.

Definition 5.1. *For $d < n$, let us say that a set $W \subset \mathbb{R}^d$ of size n is* in a forest-general position *if for any forest F such that $V(F) \subset W$ and $|E(F)|$ plus the number of isolated vertices in F is exactly d, the set*

$$\operatorname{vect} F := \{u + v \mid uv \in E(F)\} \cup \{w \in V(F) \mid w \text{ is isolated in } F\},$$

is linearly independent and of size d.

Note that this definition extends the common notion of vectors in a general linear position, which requires every d vectors to be linearly independent, since F can also be an empty forest on d vertices. The next claim extends the behavior in Definition 5.1 to forests of any size.

Claim 4. *If a set $W \subset \mathbb{R}^d$ is in a forest-general position, for any forest F such that $V(F) \subset W$,*

$$\operatorname{rank}(\operatorname{vect}(F)) = \min(d, |\operatorname{vect}(F)|).$$

Proof. If $|\operatorname{vect}(F)| = d$, the claim is by definition.

If $|\operatorname{vect}(F)| > d$, obtain a subforest F' of F such that $|\operatorname{vect}(F') = d|$. This is always possible since removing either an isolated vertex from F or a leaf of a tree in F together with the incident edge decreases the size of $\operatorname{vect}(F)$ by one. By Definition 5.1, $\operatorname{vect} F'$ is linearly independent, and since $\operatorname{vect}(F') \subset \operatorname{vect}(F)$, they both have rank d.

If $|\operatorname{vect}(F)| < d$, obtain F' by adding isolated vertices or edged in F such that $\operatorname{vect}(F') = d$. This is always possible since $d < n$. By Definition 5.1, $\operatorname{vect}(F')$ is linearly independent, and since $\operatorname{vect}(F) \subset \operatorname{vect}(F')$, $\operatorname{vect}(F)$ has full rank. □

Finally, to use Definition 5.1 in our reduction, we need to construct a corresponding set of points.

Claim 5. *For any n and d, there exists a set of n vectors in $\mathbb{N}^d$ in a forest-general position such that the total bit-length of the coordinates is bounded by $\operatorname{poly}(n + d)$.*

Proof. Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of n vectors in $\mathbb{R}^d$. Assume that Definition 5.1 does not hold for W and a particular forest F. Then there exists a linear combination of $\operatorname{vect}(F)$ which is a zero vector. Without loss of generality, $V(F) = \{w_1, w_2, \cdots, w_t\}$, let $\mathbf{a} = (a_i)_{i=1}^t$ be the zero linear combination of $\operatorname{vect}(F)$, treated as a linear combination of vectors in $V(F)$. We claim that $\mathbf{M} \cdot \mathbf{a}$ is a zero vector for the $t \times t$ matrix $\mathbf{M}$ defined as follows. The first d rows are the coordinates of w_1, w_2, …, w_t, written as columns. Since $\mathbf{a}$ is the zero linear combination of these vectors, clearly each of these rows times $\mathbf{a}$ is zero. Next, for every connected component C of F append a row $\mathbf{r}$ such that $\mathbf{r}_i = 0$ if $w_i \notin C$, and $\mathbf{r}_i = \pm 1$ if $w_i \in C$, and for every edge of C its endpoints have different signs, this encodes a 2-coloring of C. The orthogonality to $\mathbf{a}$ follows from observing how $\mathbf{a}$ on the coordinates of C is obtained from the original linear combination of $\operatorname{vect}(F)$. Note that now we have exactly t rows. Thus, Definition 5.1 does not hold if and only if $\mathbf{M}$ is non-invertible.

Now, the way to construct the required set is to take the rows of a Vandermonde matrix where the generating elements are selected to be in a certain general position regarding the differences between them, in this case it is possible to show that every matrix of the form discussed above is invertible. For the sake of brevity we omit this technical argument, and instead explain the simple randomized procedure of generating the set, which also shows why sets in a forest-general position are common.

Let W be a set of n vectors in $\mathbb{N}^d$ where each coordinate of each vector is sampled independently and uniformly from the set $\{1, \cdots, N\}$, where N is a value we fix later. The number of matrices which must be invertible by the observation above is at most $d \cdot \binom{n}{2d} \cdot d^{2d} \cdot 2^{2d}$, since the matrix up to permutations is defined by the number of components in F (at most d), the choice of t vectors in W (at most $\binom{n}{2d}$), the partition of them into components (at most d^{2d}), and the 2-coloring on each component (at most 2^{2d}). We bound the probability that a fixed matrix is non-invertible, and then apply union bound. For a fixed matrix $\mathbf{M}$, we can treat the process as follows. First we are given the $t-d$ rows obtained from the components of the forest, and then we sample one by one the d rows each composed out of t coordinates of vectors in W. For the first row, probability of falling into the span of already existing rows is at most $1/N^d$, for the second it is $1/N^{d-1}$, and so on. Then the probability of success for $\mathbf{M}$ is at least $(1-1/N) \cdot (1-1/N^2) \cdots$, which could be lower-bounded by $1 - 1/(N-1)$. By taking N sufficiently large compared to the number of matrices, the union bound is satisfied. Note that $\log N$ is polynomially

bounded in n and d, and thus the total bit-length is also polynomially bounded. □

Now we are ready to show the reduction. Assume that we are given an instance (G, r) of CLIQUE with $|V(G)| = n$, $|E(G)| = m$. Set $d = r + 1$, one larger than the size of the clique to find. By Claim 5, obtain a set $W = \{w_1, \ldots, w_n\}$ of n vectors in $\mathbb{R}^d$ in a forest-general position.

Vectors from W are associated with vertices of G. Now consider the matrix $\mathbf{A}$ where rows correspond to the edges of G,

$$\mathbf{A} = (w_i + w_j \mid \{v_i, v_j\} \in E(G)) .$$

The matrix $\mathbf{A}$ is the input matrix to ROBUST SUBSPACE RECOVERY, the target rank is r, the number of outliers k is set to $m - \binom{r}{2}$. We claim that (G, r) is a yes-instance of CLIQUE if and only if the constructed instance of ROBUST SUBSPACE RECOVERY is a yes-instance.

To reformulate the objective of ROBUST SUBSPACE RECOVERY, an instance $(\mathbf{A}, r, k)$ is a yes-instance if and only if there is a subset of $n - k = \binom{r}{2}$ rows of $\mathbf{A}$ with rank at most r. The following claim shows how to identify the rank just by the structure of the edges corresponding to the selected rows.

Claim 6. *The rank of any submatrix $\mathbf{A}' \subset \mathbf{A}$ obtained by deleting rows from $\mathbf{A}$ is*

$$\max(d, |V(G(\mathbf{A}'))| - \text{number of bipartite graphs among } \operatorname{cc}(G(\mathbf{A}'))),$$

where $G(\mathbf{A}')$ is the subgraph of G such that its edges are exactly the edges corresponding to the rows of $\mathbf{A}'$, and its vertex set is the set of all endpoints of the edges, and $\operatorname{cc}(G(\mathbf{A}'))$ is the set of connected components of $G(\mathbf{A}')$.

Proof. We will modify $\mathbf{A}'$ to $\operatorname{vect}(F)$ for a certain forest F, by using elemental row operations which do not change rank. Then Claim 4 finishes the proof.

The modification is performed on each connected component of $G(\mathbf{A}')$ independently. Consider a set of rows $\mathbf{C}$ which corresponds to the edges of a connected component in $G(\mathbf{A}')$. If $G(M')$ contains an odd cycle, then from the rows of $\mathbf{C}$ we can obtain all the underlying elements of W by elimination: start from the row corresponding to one edge of a cycle, iteratively subtract/add subsequent edges of the cycle. In the end we are left with $2w$, where w is a vector in W which corresponds to the vertex of the cycle we started the elimination from. After multiplication by $1/2$ we have one of the vertex vectors. Now by consequently subtracting from the edge vectors along a spanning tree of $G(\mathbf{C})$ we can obtain all the vertex vectors of $V(G(\mathbf{C}))$. After zeroing out the remaining edge vectors $\mathbf{C}$ is $(w_i | v_i \in V(G(\mathbf{C})))$, up to permuting the rows and appending zero rows.

In the other case, if $G(\mathbf{C})$ is bipartite, consider the matrix $\mathbf{S} \subset \mathbf{C}$ which corresponds to the edges of a spanning tree T of $G(\mathbf{C})$. Let uv be an edge of $G(\mathbf{C})$ which is not in T. Since $G(\mathbf{C})$ is bipartite, u and v are connected by a path in T with odd number of edges. Then by consequently adding/subtracting the edge vectors of this path we can obtain the edge vector corresponding to uv, in the same fashion as in the previous case. Thus we can zero out all rows of $\mathbf{C}$ except for $\mathbf{S}$.

After dealing with each component, consider the forest F where $V(F) = V(G(\mathbf{A}'))$, and $E(F)$ is the union of the edges of all spanning trees picked during the modification in the bipartite components. Thus, vertices of all the non-bipartite components are isolated in F. We claim that the set of non-zero rows of matrix $\mathbf{B}$ obtained from $\mathbf{A}'$ by the modifications above is exactly $\operatorname{vect}(F)$. Indeed, for each non-bipartite component we obtained all the vertex vectors while zeroing out everything else, and for each bipartite component we kept only the edge vectors of the corresponding spanning tree. Now, since modifications are rank-preserving,

$$\begin{aligned}\operatorname{rank}(\mathbf{A}') = \operatorname{rank}(\mathbf{B}) &= \operatorname{rank}(\operatorname{vect}(F)) \\ &= \min(d, |\operatorname{vect}(F)|)\end{aligned}$$

by Claim 4. The size of $\operatorname{vect}(F)$ by definition is the number of isolated vertices in F plus the number of edges in F, and that is equal to the number of vertices in F minus the number of non-trivial connected components in F, since F is a forest. Finally, we observe that $|V(F)| = |V(G(\mathbf{A}'))|$, and bipartite components of $G(\mathbf{A}')$ are in one-to-one correspondence with non-empty components of F, finishing the proof of Claim 6.

□

With Claim 6 proven, we show that the only way to keep at least $\binom{r}{2}$ edge vectors in such a way that the rank is at most r, is to select the rows corresponding to the edges of an r-clique. For any $\mathbf{A}' \subset \mathbf{A}$, let $\kappa(\mathbf{A}') = \frac{|\mathbf{A}'|}{\operatorname{rank} \mathbf{A}'}$, the number of edges per unit of rank. We claim that κ is strictly maximized on an r-clique over all $\mathbf{A}' \subset \mathbf{A}$ which have rank at most r.

For a matrix $\mathbf{K}$ corresponding to an r-clique, $\kappa(\mathbf{K}) = \frac{r-1}{2}$ by Claim 6. Consider any $\mathbf{A}' \subset \mathbf{A}$ such that $G(\mathbf{A}')$ is connected. Since $\operatorname{rank}(\mathbf{A}')$ is at most r, there are two possibilities by Claim 6. If $G(\mathbf{A}')$ is a non-bipartite graph on at most r vertices, it has less edges than an r-clique, and so $\kappa(\mathbf{A}') < \frac{r-1}{2}$. Otherwise $G(\mathbf{A}')$ is bipartite, and $|V(G(\mathbf{A}'))|$ must be $r + 1$. Then there are at most $\left(\frac{r+1}{2}\right)^2$ edges, so $\kappa(\mathbf{A}') < \frac{r-1}{2}$ for $r \geq 4$.

To prove the statement for any $\mathbf{A}' \subset \mathbf{A}$ such that $\operatorname{rank}(\mathbf{A}') \leq r$, we do an induction on the number of con-

nected components in $G(\mathbf{A}')$. The base case when there is only one connected component is already proven. Now, consider $\mathbf{A}' = \mathbf{B} \cup \mathbf{C}$ where $G(\mathbf{C})$ is connected. By Claim 6, $\operatorname{rank}(\mathbf{B} \cup \mathbf{C}) = \operatorname{rank}(\mathbf{B}) + \operatorname{rank}(\mathbf{C})$, and also $|\mathbf{A}'| = |\mathbf{B}| + |\mathbf{C}|$. By the induction, $|\mathbf{B}|/\operatorname{rank}(\mathbf{B}) < \frac{r-1}{2}$, and $|\mathbf{C}|/\operatorname{rank}(\mathbf{C}) < \frac{r-1}{2}$, so

$$\kappa(\mathbf{A}') = \frac{|\mathbf{B}| + |\mathbf{C}|}{\operatorname{rank}(\mathbf{B}) + \operatorname{rank}(\mathbf{C})} < \frac{r-1}{2}.$$

Thus, the rows selected have rank r if and only if they correspond to an r-clique, which proves the correctness of the reduction.

By (Chen et al., 2006), see also (Cygan et al., 2015), assuming ETH, there is no algorithm solving CLIQUE in time $f(r) \cdot n^{o(r)}$ where r is the size of the clique, for any computable function f of r. Since in our reduction $d = r + 1$, the theorem follows.[2] □

Finally we note that since ROBUST SUBSPACE RECOVERY is the zero-valued restriction of PCA WITH OUTLIERS, the hardness of approximation for the latter easily follows from Theorem 1.5.

Corollary 5.1. *Assuming ETH, there is no algorithm approximating* PCA WITH OUTLIERS *with any multiplicative guarantee in time* $f(d) \cdot n^{o(d)}$.

Proof. An algorithm described in the statement could distinguish between $OPT \leq D$ and $OPT > \alpha \cdot D$ for given D, where α is the approximation guarantee which may depend on the input instance. Then this algorithm could also distinguish between $OPT = 0$ and $OPT > 0$, violating Theorem 1.5. □

5.2. Proof of Theorem 1.1

We restate the theorem here for convenience.

Theorem 1.1. *For every $\varepsilon > 0$, an $(1+\varepsilon)$-approximate solution to* PCA WITH OUTLIERS *can be found in time* $n^{\mathcal{O}(\frac{r \log r}{\varepsilon^2})} \cdot d^{\mathcal{O}(1)}$.

First, we recall the known results about column sampling. We will use the ridge leverage score construction due to (Cohen et al., 2017). For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and an index $i \in [n]$ the *i-th ridge leverage score of* $\mathbf{A}$ is given as

$$\tau_i(A) = \mathbf{a_{i:}}(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^+\mathbf{a_{i:}}^T,$$

where $\lambda = ||\mathbf{A} - \mathbf{A}_r||_F^2/k$, and $+$ denotes the Moore-Penrose pseudoinverse of a matrix. The following statement about sampling w.r.t. to ridge leverage scores is proven in (Cohen et al., 2017).

[2] The same reduction also shows that ROBUST SUBSPACE RECOVERY is W[1]-hard when parameterized by $(d, n-k)$.

Theorem 5.2 (Theorem 6 in (Cohen et al., 2017))**.** *For $i \in [n]$, let $\tilde{\tau}_i \geq \tau_i(\mathbf{A})$ be an overestimate for the i-th ridge leverage score. Let $p_i = \frac{\tilde{\tau}_i}{\sum_i \tilde{\tau}_i}$. Let $t = \frac{c \log(r/\delta)}{\varepsilon^2} \sum_i \tilde{\tau}_i$ for any $\varepsilon > 0$ and some sufficiently large constant c. Construct $\mathbf{C}$ by sampling t rows of $\mathbf{A}$, each set to $\frac{1}{\sqrt{tp_i}}\mathbf{a_{i:}}$ with probability p_i. With probability $1 - \delta$, for any rank r orthogonal projection $\mathbf{X}$,*

$$(1-\varepsilon)||\mathbf{A}-\mathbf{A}\mathbf{X}||_F^2 \leq ||\mathbf{C}-\mathbf{C}\mathbf{X}||_F^2 \leq (1+\varepsilon)||\mathbf{A}-\mathbf{A}\mathbf{X}||_F^2.$$

Our objective will be to guess certain ridge leverage score overestimates. For that, we will need a lemma bounding the range of ridge leverage scores, following from (Cohen et al., 2017).

Lemma 5.3.

$$\frac{1}{2} \leq \sum_{i=1}^{n} \tau_i(\mathbf{A}) \leq 2r.$$

Proof. The upper bound is precisely given by Lemma 4 in (Cohen et al., 2017). For the lower bound,

$$\begin{aligned}\sum_{i=1}^{n} \tau_i(\mathbf{A}) &= \sum_{i=1}^{n} \frac{\sigma_i(\mathbf{A})^2}{\sigma_i(\mathbf{A})^2 + \frac{||\mathbf{A}-\mathbf{A}_r||_F^2}{r}} \\ &= \sum_{i=1}^{n} \frac{\sigma_i(\mathbf{A})^2}{\sigma_i(\mathbf{A})^2 + \frac{1}{r}\sum_{j=r+1}^{n} \sigma_j(\mathbf{A})^2} \\ &\geq \sum_{i=r+1}^{n} \frac{\sigma_i(\mathbf{A})^2}{\left(1+\frac{1}{r}\right)\sum_{j=r+1}^{n} \sigma_j(\mathbf{A})^2} \geq \frac{1}{1+\frac{1}{r}} \geq \frac{1}{2},\end{aligned}$$

where the first equality is given by the proof of Lemma 4 in (Cohen et al., 2017), and then we lower bound the first r terms of the sum by zero. □

Now we are ready to prove the main result of this section.

Proof of Theorem 1.1. The algorithm proceeds as follows. Set $T = 6r$, a sufficiently small $\varepsilon_0 < \varepsilon$ (to be defined later), and $t = \frac{c \log(2r)}{\varepsilon_0^2} T$ where c is a sufficiently large constant from the statement of Theorem 5.2. First, guess t indices $\{i_1, \ldots i_t\}$ in $[n]$, each corresponding to a row in $\mathbf{A}$. For each index i in $\{i_1, \ldots, i_t\}$, guess a value $\tilde{\tau}_i$ from the set $\mathcal{T} = \{\frac{2r}{2^q}, \frac{2r}{2^{p-1}}, \ldots, 2r\}$, where q is the smallest integer such that $2^q \geq n$. Compose the matrix $\mathbf{C} \in \mathbb{R}^{t \times d}$ from the rows of A: for each $j \in [t]$, take the i_j-th row of $\mathbf{A}$ multiplied by $\frac{1}{\sqrt{t\tilde{\tau}_i}}$. By using the standard PCA algorithm, find in polynomial time the optimal low-rank approximation of $\mathbf{C}$, i.e. the rank r orthogonal projection matrix $\mathbf{X} \in \mathbb{R}^{d \times d}$ minimizing $||\mathbf{C} - \mathbf{C}\mathbf{X}||_F^2$. Construct the matrix $\mathbf{N} \in \mathbb{R}^{n \times d}$ such that it contains k rows of $\mathbf{A}$ maximizing the distance to the r-dimensional subspace corresponding to $\mathbf{X}$, at the

respective positions of these rows in $\mathbf{A}$, and all the other rows of $\mathbf{N}$ are zero rows. Set $\mathbf{L}$ to be $(\mathbf{A}-\mathbf{N})\mathbf{X}$. Finally, return $\mathbf{L}$ and $\mathbf{N}$ minimizing the value $||\mathbf{A}-\mathbf{L}-\mathbf{N}||_F^2$ over all guesses performed by the algorithm.

Correctness of the algorithm. Clearly, the matrices $\mathbf{L}$ and $\mathbf{N}$ returned by the algorithm are subject to the constraints, that $\operatorname{rank}\mathbf{L} \leq r$ and $\mathbf{N}$ contains at most k non-zero rows, thus it only remains to prove that the cost of the returned solution is at most $(1+\varepsilon)$ times the cost of the optimal solution. Fix an optimal solution $(\mathbf{L}^*, \mathbf{N}^*)$. Denote by $\mathbf{A}^* = \mathbf{A} - \mathbf{N}^*$ the optimal inlier matrix, that is, the matrix $\mathbf{A}$ where the outlier rows are replaced by zero rows. Recall that for $i \in [n]$, $\tau_i(\mathbf{A}^*)$ is the i-th ridge leverage score of $\mathbf{A}^*$. For each $i \in [n]$, denote by $\tilde{\tau}_i^*$ the smallest element of $\mathcal{T}$ that is at least $\tau_i(\mathbf{A}^*)$, $\tilde{\tau}_i^*$ is well-defined since $\tau_i(\mathbf{A}^*)$ is at most $2r$ by Lemma 5.3, and the set $\mathcal{T}$ contains $2r$. We show now that $\sum_{i=1}^n \tilde{\tau}_i$ is at most T, and thus the number of the sampled rows $t = \frac{c\log(r/\delta)}{\varepsilon_0^2}T$ is sufficiently large to apply Theorem 5.2. For each $i \in [n]$ consider two cases. First, if $\tau_i(\mathbf{A}^*)$ is at least the smallest element of $\mathcal{T}'$, then $\tilde{\tau}_i \leq 2\tau_i(\mathbf{A}^*)$, as elements of the set $\mathcal{T}'$ are at factor two from each other. Over all such indices, $\sum_i \tilde{\tau}_i \leq 2\sum_{i=1}^n \tau_i(\mathbf{A}^*) \leq 4r$ by Lemma 5.3. Second, if $\tau_i(\mathbf{A}^*)$ is less than $\frac{2r}{2^q}$, then $\tilde{\tau}_i$ is set to this value, and the sum of all such $\tilde{\tau}_i$ is at most $2r$ as there are $n \leq 2^q$ values in total. Summing the bounds for both cases, we get that $\sum_{i=1}^n \tilde{\tau}_i \leq 6r = T$.

Now denote $\delta = 1/2$ and invoke Theorem 5.2 for $\mathbf{A}^*$ with the set values of δ, t, and $\tilde{\tau}_i$ for each $i \in [n]$, and with the error parameter ε_0. With probability $1-\delta$, the sampling procedure described in the statement of Theorem 5.2 succeeds. Since this probability is positive, there exists a particular selection of t rows that produces the desired matrix. Thus the matrix $\mathbf{C}^*$ composed of these rows and reweighted according to Theorem 5.2, satisfies

$$\begin{aligned}(1-\varepsilon_0)||\mathbf{A}^* - \mathbf{A}^*\mathbf{X}||_F^2 &\leq ||\mathbf{C}^* - \mathbf{C}^*\mathbf{X}||_F^2 \\ &\leq (1+\varepsilon_0)||\mathbf{A}^* - \mathbf{A}^*\mathbf{X}||_F^2,\end{aligned} \tag{3}$$

for any rank r orthogonal projection matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$. Denote the indices of these rows by $i_1^*, \ldots, i_t^*$. In one of the branches, our algorithm considers the values $i_1 = i_1^*, \ldots, i_t = i_t^*$, and $\tilde{\tau}_i = \tilde{\tau}_i^*$ for all $i \in \{i_1, \cdots, i_t\}$. Thus the matrix $\mathbf{C}$ constructed by our algorithm at this step is exactly the matrix $\mathbf{C}^*$ where every entry is multiplied by $1/\sqrt{\sum_{i=1}^n \tilde{\tau}_i}$. Consider the orthogonal projection matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$ that provides the optimal rank k approximation of $\mathbf{C}$, and also of $\mathbf{C}^*$ since these two matrices are identical up to multiplying by a constant. Let $\mathbf{X}^*$ be the projection matrix of the optimal solution, that is, $\mathbf{L}^* = \mathbf{A}^*\mathbf{X}^*$. Then by (3), and because $\mathbf{X}$ gives the best low-rank approximation for $\mathbf{C}^*$, we have that

$$\begin{aligned}||\mathbf{A}^* - \mathbf{A}^*\mathbf{X}||_F^2 &\leq \frac{1}{1-\varepsilon_0}||\mathbf{C}^* - \mathbf{C}^*\mathbf{X}||_F^2 \\ &\leq \frac{1}{1-\varepsilon_0}||\mathbf{C}^* - \mathbf{C}^*\mathbf{X}^*||_F^2 \\ &\leq \frac{1+\varepsilon_0}{1-\varepsilon_0}||\mathbf{A}^* - \mathbf{A}^*\mathbf{X}^*||_F^2 \leq \frac{1+\varepsilon_0}{1-\varepsilon_0}OPT.\end{aligned} \tag{4}$$

Finally, denote $\mathbf{A}' = \mathbf{A} - \mathbf{N}$. Let us note that both $\mathbf{A}'$ and $\mathbf{A}^*$ contain certain $(n-k)$ rows of $\mathbf{A}$, but $\mathbf{A}'$ contains precisely the $(n-k)$ rows that incur the smallest loss w.r.t. $\mathbf{X}$. Therefore,

$$||\mathbf{A}' - \mathbf{A}'\mathbf{X}||_F^2 \leq ||\mathbf{A}^* - \mathbf{A}^*\mathbf{X}||_F^2. \tag{5}$$

Since $\mathbf{L}$ is exactly $\mathbf{A}'\mathbf{X}$, by (4) and (5), we have

$$||\mathbf{A} - \mathbf{L} - \mathbf{N}||_F^2 \leq \frac{1+\varepsilon_0}{1-\varepsilon_0}OPT.$$

Setting $\varepsilon_0 = \Theta(\varepsilon)$ so that $\frac{1+\varepsilon_0}{1-\varepsilon_0}$ is at most $1+\varepsilon$, concludes the proof of correctness.

Running time. The algorithm consider n choices for each of the t values $i_1, \ldots, i_t$, and $\mathcal{O}(\log n)$ choices for each of the t values $\tilde{\tau}_{i_1}, \ldots, \tilde{\tau}_{i_t}$, where $t = O(r\log r/\varepsilon^2)$. For each choice, the optimal low-rank approximation and the outliers are found in polynomial time. Thus, the total running time of the algorithm is upper-bounded by

$$(n\log n)^{\mathcal{O}(t)}\operatorname{poly}(nd) = n^{\mathcal{O}(r\log r/\varepsilon^2)}\operatorname{poly}(d).$$

□

5.3. Proof of Theorem 1.4

In this section we prove Theorem 1.4 that Robust Subspace Recovery is solvable in time $2^{\mathcal{O}(\min\{k, r\log r\}\cdot(\log r+\log k))}\cdot(nd)^{\mathcal{O}(1)}$. The algorithm we give is almost identical to the algorithm of (Fomin et al., 2018b) for the Matrix Rigidity problem. We provide here full details for completeness.

Let us remind that in Robust Subspace Recovery, we are given a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, whose rows correspond to the data points, and integers r, k. The question is whether there are matrices $\mathbf{L}$ and $\mathbf{N}$ such that $\mathbf{A} = \mathbf{L} + \mathbf{N}$, the rank of $\mathbf{L}$ is at most r, and $\mathbf{N}$ has at most k non-zero rows. Equivalently, the question is whether it is possible to delete at most k rows of $\mathbf{A}$ such that the resulting matrix is of rank at most r.

We need the following observation: For every set X of $r+1$ independent rows of matrix $\mathbf{A}$, at least one row from X is an outlier. In other words,

Proposition 5.4. *Let X be a set of indices of $r+1$ independent rows of $\mathbf{A}$. Then for every optimal solution $(\mathbf{L}, \mathbf{N})$, at least one index from X is the index of a non-zero row of $\mathbf{N}$.*

Proof. The rank of $\mathbf{L}$ is at most r, thus $\mathbf{L}$ cannot contain more than r independent rows. □

The crux of the algorithm is in the procedure that for an input $(\mathbf{A}, r, k)$ of ROBUST SUBSPACE RECOVERY in polynomial time constructs an equivalent instance $(\tilde{\mathbf{A}}, r, k)$, with matrix $\tilde{\mathbf{A}}$ containing at most $(r+1)(k+1)$ rows. We assume that the rank of $\mathbf{A}$ is more than r, otherwise $\mathbf{L} = \mathbf{A}$ is trivially a solution. We also assume that $n > (r+1)(k+1)$ because otherwise we can put $\tilde{\mathbf{A}} = \mathbf{A}$. The procedure runs in two steps.

First, we find pairwise disjoint sets $R_1, \ldots, R_t$ of rows in $\mathbf{A}$. Each set R_i consists of $r+1$ independent rows. We construct such sets greedily by picking a set of $r+1$ independent rows and deleting them from $\mathbf{A}$ until the rank of the remaining rows will be at most r. By Proposition 5.4, each of these sets R_i contains at least one outlier. Thus if $t > k$, the rank of $\mathbf{A}$ cannot be reduced to r by deleting k rows, hence $(\mathbf{A}, r, k)$ is a no-instance.

Second, from the remaining rows of $\mathbf{A}$, that is, the rows that are not in $R_1 \cup \cdots \cup R_t$, we select pairwise disjoint sets $R_{t+1}, \ldots, R_{k+1}$ as follows. For $i \geq t+1$ let M_i be the rows of $\mathbf{A}$ that are not in $R_1 \cup \cdots \cup R_{i-1}$. Then R_i is a subset of M_i forming its basis. Note that $|R_i| \leq r$ for $i \geq t+1$.

Finally, the matrix $\tilde{\mathbf{A}}$ is the matrix whose rows are $R_1 \cup \cdots \cup R_{k+1}$. At every step of the construction of $\tilde{\mathbf{A}}$ we find an independent set of rows and thus the total running time is polynomial. Since every set R_i contains at most $r+1$ row, matrix $\tilde{\mathbf{A}}$ contains at most $(r+1) \cdot (k+1)$ rows. Thus what remains is to show the equivalence of both instances.

Lemma 5.5. *$(\mathbf{A}, r, k)$ is a yes-instance of* ROBUST SUBSPACE RECOVERY *if and only if $(\tilde{\mathbf{A}}, r, k)$ is a yes-instance.*

Proof. In one direction the proof is trivial. If the rank of $\mathbf{A}$ can be reduced to r by deleting at most k rows, the same is true for $\tilde{\mathbf{A}}$.

For the opposite direction. Let O be the set of outliers, that is, the set of rows of $\tilde{\mathbf{A}}$ of size at most k whose removal decreases the rank of the matrix down to r. The rows of matrix $\mathbf{A}$ are partitioned into the rows of $\tilde{\mathbf{A}}$ and the remaining rows, which we denote by M. (By slightly abusing notation, we do not distinguish a matrix and a set of rows forming this matrix.) We claim that removing rows of O from $\mathbf{A}$ also reduces its rank to r. Targeting a contradiction, let us assume that this is not true. Then $\mathbf{A} \setminus O$ contains a set X of $r+1$ linearly independent rows.

The rows of $\tilde{\mathbf{A}}$ are $R_1 \cup \cdots \cup R_t \cup \cdots \cup R_{k+1}$. Because $|O| \leq k$, by the pigeonhole principle, there is R_i that does not contain a row from O. For $i \leq t$, each R_i consists of $r+1$ independent rows, and by Proposition 5.4 must contain at least one row from O. This means that at least one R_i, $i > t$, contains no row from O. But by the construction of sets R_i for $i > t$, the set of rows M is in the span of R_i. Hence the set $X' = (X \cap \tilde{\mathbf{A}}) \cup R_i \subseteq \tilde{\mathbf{A}} \setminus O$ contains $r+1$ linearly independent rows of $\tilde{\mathbf{A}} \setminus O$, which is a contradiction. □

Finally, to prove Theorem 1.4, for input $(\mathbf{A}, r, k)$, we construct an equivalent instance $(\tilde{\mathbf{A}}, r, k)$, where $\tilde{\mathbf{A}}$ contains at most $(r+1)(k+1)$ rows. For each subset of rows of $\tilde{\mathbf{A}}$ of size k, we check whether removal of this set results in a matrix of rank at most r. If we found such a set O, by Lemma 5.5, the same rows are outliers for $\mathbf{A}$ as well. If we did not find a set of outliers for $\tilde{\mathbf{A}}$, we can safely conclude that $(\mathbf{A}, r, k)$ is a no-instance. Construction of the reduced instance can be done in polynomial time, and the number of all subsets of rows of $\tilde{\mathbf{A}}$ of size k, does not exceed $\binom{(r+1)(k+1)}{k} = 2^{\mathcal{O}(k(\log r + \log k))}$. Hence the total running time is $2^{\mathcal{O}(k(\log r + \log k))} \cdot (nd)^{\mathcal{O}(1)}$.

Alternatively, instead of trying all subsets of k rows of $\tilde{\mathbf{A}}$, we can run the algorithm of Theorem 1.1 on the instance $(\tilde{\mathbf{A}}, r, k)$ with the error parameter ε set to an arbitrary constant. Recall that by Theorem 1.1 an $(1+\varepsilon)$-approximate solution to PCA WITH OUTLIERS can be found in time $n^{\mathcal{O}(\frac{r \log r}{\varepsilon^2})} \cdot d^{\mathcal{O}(1)}$. Since an instance of ROBUST SUBSPACE RECOVERY is a yes-instance if and only if it has the objective value of zero as an instance of PCA WITH OUTLIERS, a constant-factor approximation for PCA WITH OUTLIERS suffices for solving ROBUST SUBSPACE RECOVERY exactly. Thus the whole algorithm for ROBUST SUBSPACE RECOVERY finishes in time $2^{\mathcal{O}(r \log r (\log r + \log k))}(nd)^{\mathcal{O}(1)}$, as there are at most $(r+1)(k+1)$ rows in the matrix $\tilde{\mathbf{A}}$. The running time of $2^{\mathcal{O}(\min\{k, r \log r\} \cdot (\log r + \log k))} \cdot (nd)^{\mathcal{O}(1)}$ follows by combining the above two algorithms.

5.4. Proof of Theorem 2.2

Let (a) $\tilde{\mathbf{x}}_i = \arg\min_{\mathbf{x}} \left\|\mathbf{a^T S} - \mathbf{x^T V S}\right\|_2^2$ and (b) $\hat{\mathbf{x}}_i = \arg\min_{\mathbf{x}} \left\|\mathbf{a^T} - \mathbf{x^T V}\right\|_2^2$. Since $\mathbf{S}$ is a ε-embedding for $\mathrm{col}([\mathbf{V^T}|\mathbf{a}])$, we have

$$\begin{aligned}
(1-\varepsilon)\,\mathrm{dist}^2(\mathbf{a^T}, \mathbf{V}) &= (1-\varepsilon)\left\|\mathbf{a^T} - \hat{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{V}\right\|_2^2 \\
&\leq (1-\varepsilon)\left\|\mathbf{a^T} - \tilde{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{V}\right\|_2^2 \quad \text{(By } (b)\text{)} \\
&\leq \left\|\mathbf{a^T S} - \tilde{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{V S}\right\|_2^2 \\
&= \mathrm{dist}^2(\mathbf{a^T S}, \mathbf{V S}).
\end{aligned}$$

Similarly, we have

$$\mathrm{dist}^2(\mathbf{a^T S}, \mathbf{V S}) = \left\|\mathbf{a^T S} - \tilde{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{V S}\right\|_2^2$$

$$\leq \left\|\mathbf{a^T S} - \hat{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{VS}\right\|_2^2 \quad (\text{By } (a))$$
$$\leq (1+\varepsilon)\left\|\mathbf{a^T} - \hat{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{V}\right\|_2^2$$
$$= (1+\varepsilon)\,\mathrm{dist}^2(\mathbf{a^T}, \mathbf{V}).$$

which gives us

$$(1-\varepsilon)\,\mathrm{dist}^2(\mathbf{a^T}, \mathbf{V}) \leq \mathrm{dist}^2(\mathbf{a^T S}, \mathbf{VS})$$
$$\leq (1+\varepsilon)\,\mathrm{dist}^2(\mathbf{a^T}, \mathbf{V}).$$

5.5. Proof of Theorem 2.3

First we state the following theorem from (Clarkson & Woodruff, 2013), see Theorem 39, which gives sufficient conditions for $\mathbf{S}$ to be a ε-affine embedding

Theorem 5.6. *Let $\mathbf{U} \in \mathbb{R}^{n\times r}$ and $\mathbf{B} \in \mathbb{R}^{n\times d'}$, then $\mathbf{S} \in \mathbb{R}^{s\times n}$ is a 3ε-affine embedding for $(\mathbf{U}, \mathbf{B})$ if the following event occurs*

1. *Subspace Embedding: $\mathbf{S}$ is a subspace embedding for column space of $\mathbf{U}$.*
2. *Approximate Matrix Multiplication: For arbitrary fixed matrices $\mathbf{A}$ and $\mathbf{B}$ of n rows we have*

$$\left\|\mathbf{A^T S^T SB} - \mathbf{A^T B}\right\|_F^2 \leq \frac{\varepsilon^2}{r}\left\|\mathbf{A}\right\|_F^2\left\|\mathbf{B}\right\|_F^2$$

3. *Preserve Matrix Norm: For arbitrary fixed matrix $\mathbf{A}$ of n rows $\left\|\mathbf{SA}\right\|_F^2 = (1\pm\varepsilon)\left\|\mathbf{A}\right\|_F^2$*

The next two lemmas show that random Gaussian matrices satisfy condition (2) and (3) of above theorem.

Lemma 5.7. *Let $0 < \varepsilon, \delta < 1$ and $\mathbf{S} = \frac{1}{\sqrt{s}}\mathbf{G} \in \mathbb{R}^{s\times n}$ where the entries of matrix $\mathbf{G}$ are independent standard normal random variables. Then for $s = \mathcal{O}(\frac{1}{\varepsilon^2}\log(1/\delta))$*

$$Pr_\mathbf{S}(\left\|\mathbf{A^T S^T SB} - \mathbf{A^T B}\right\|_F^2 \leq \varepsilon^2\left\|\mathbf{A}\right\|_F^2\left\|\mathbf{B}\right\|_F^2) \geq 1-\delta$$

Proof. Proof follows from Theorem 6.2 and Remark 6.3 from (Kane & Nelson, 2014). □

For the next lemma, we use following inequality form (Hanson & Wright, 1971)

Theorem 5.8. *(Hanson-Wright Inequality) Let $\mathbf{g} \in \mathbb{R}^n$ be a vector of standard normal random variables and $\mathbf{A} \in \mathbb{R}^{n\times n}$ then there exist a constant $C > 0$ such that for all $\varepsilon > 0$*

$$Pr_\mathbf{g}(|\mathbf{g^T Ag} - E[\mathbf{g^T Ag}]| > \varepsilon) \leq e^{-C\varepsilon^2/\left\|\mathbf{A}\right\|_F^2} + e^{-C\varepsilon/\left\|\mathbf{A}\right\|_2}$$

Lemma 5.9. *Let $0 < \varepsilon, \delta < 1$ and $\mathbf{S} = \frac{1}{\sqrt{s}}\mathbf{G} \in \mathbb{R}^{s\times n}$ where the entries of matrix $\mathbf{G}$ are independent standard normal random variables. Then for $s = \mathcal{O}(\frac{1}{\varepsilon^2}\log(1/\delta))$, for any fixed arbitrary matrix $\mathbf{A}$ of n rows*

$$Pr_\mathbf{S}(|\left\|\mathbf{SA}\right\|_F^2 - \left\|\mathbf{A}\right\|_F^2| \leq \varepsilon\left\|\mathbf{A}\right\|_F^2) \geq 1-\delta$$

Proof. First we show that $E[\left\|\mathbf{SA}\right\|_F^2] = \left\|\mathbf{A}\right\|_F^2$.

$$E_\mathbf{S}[\left\|\mathbf{SA}\right\|_F^2] = E_\mathbf{S}[tr(\mathbf{A^T S^T SA})]$$
$$= tr(\mathbf{A^T}E_\mathbf{S}[\mathbf{S^T S}]\mathbf{A})$$
$$= tr(\mathbf{A^T I_n A}) = \left\|\mathbf{A}\right\|_F^2$$

Now let $\mathbf{g_i} \in \mathbb{R}^n$ denote the i-th row of $\mathbf{G}$. Let $\mathbf{X_i} = \mathbf{g_i^T AA^T g_i}$ and $\mathbf{X} = \frac{1}{s}\sum_{i=1}^s \mathbf{X_i}$. Then observe that

$$\left\|\mathbf{SA}\right\|_F^2 = \frac{1}{s}\sum_{i=1}^s \mathbf{g_i^T AA^T g_i} = \frac{1}{s}\sum_{i=1}^s \mathbf{X_i} = \mathbf{X}$$

Using Hanson-Wright Inequality we have that for $\varepsilon < 1$

$$Pr_{\mathbf{g_i}}(|\mathbf{g_i^T AA^T g_i} - E[\mathbf{g_i^T AA^T g_i}]| > \varepsilon\left\|\mathbf{A}\right\|_F^2)$$
$$\leq e^{-C\varepsilon^2} + e^{-C\varepsilon} \leq 2e^{-C\varepsilon^2}$$

Which tell us that $\mathbf{X_i}$ is a sub-Gaussian random variable and that $\mathbf{X} = \left\|\mathbf{SA}\right\|_F^2$ is an average of s independent sub-Gaussian random variables. Using Chernoff tail-inequality for sub-Gaussian random variables

$$Pr_\mathbf{S}(|\left\|\mathbf{SA}\right\|_F^2 - \left\|\mathbf{A}\right\|_F^2| > \varepsilon\left\|\mathbf{A}\right\|_F^2)$$
$$= Pr_\mathbf{S}(|\mathbf{X} - E[\mathbf{X}]| > \varepsilon\left\|\mathbf{A}\right\|_F^2) \leq e^{\mathcal{O}(-s\varepsilon^2)} \leq \delta$$

Combining the above two lemmas with Theorem 2.1 completes the proof of 2.3 □