Multi-Subspace Matrix Recovery from Permuted Data

Liangqi Xie, Jicong Fan*

School of Data Science, The Chinese University of Hong Kong, Shenzhen
222041028@link.cuhk.edu.cn, fanjicong@link.cuhk.edu.cn

Abstract

This paper aims to recover a multi-subspace matrix from permuted data: given a matrix, in which the columns are drawn from a union of low-dimensional subspaces and some columns are corrupted by permutations on their entries, recover the original matrix. The task has numerous practical applications such as data cleaning, integration, and de-anonymization, but it remains challenging and cannot be well addressed by existing techniques such as robust principal component analysis because of the presence of multiple subspaces and the permutations on the elements of vectors. To solve the challenge, we develop a novel four-stage algorithm pipeline including outlier identification, subspace reconstruction, outlier classification, and unsupervised sensing for permuted vector recovery. Particularly, we provide theoretical guarantees for the outlier classification step, ensuring reliable multi-subspace matrix recovery. Our pipeline is compared with state-of-the-art competitors on multiple benchmarks and shows superior performance.

1 Introduction

1.1 Background and Motivation

Permutation is a critical form of data corruption in many applications such as computer vision, data integration, and privacy protection, and hence therefore requires significant attention. This section highlights two key applications: record linkage and de-anonymization, related to data integration and privacy protection, respectively. In record linkage, the goal is to integrate data from different sources for analysis (Fellegi and Sunter 1969; Muralidhar 2017). Columns of a data matrix, gathered independently, may not correspond to the same entity in each row. Thus, reordering or recovering these columns is essential for accurate analysis. In de-anonymization, data providers anonymize information by shuffling the columns of the ground-truth data matrix before release. Recovering the original data becomes a reverse process of data protection (Domingo-Ferrer and Muralidhar 2016). This has practical implications, especially in healthcare and finance, where data integrity is crucial.

Typically, the ground-truth data matrix has a lower rank than the permuted version. However, our study addresses a more general scenario where the ground-truth matrix can be full-rank (Fan and Chow 2018; Fan and Udell 2019; Fan, Zhang, and Udell 2020), leading us to explore Permuted Matrix Recovery with Multi-Subspace Data.

1.2 Related Work

While several studies address label-entity mismatches, only two focus on matrix recovery. The first, (Yao, Peng, and Tsakiris 2021), combines robust PCA (Candès et al. 2011) with unlabeled sensing, estimating the ground-truth subspace first and using it to recover permuted data. The second, (Tang et al. 2021), estimates the permutation matrix within a Birkhoff polytope, minimizing an objective inspired by nuclear norm minimization. This reformulates the problem into continuous optimization within the polytope, solved via proximal gradient methods and the Sinkhorn algorithm (Cuturi 2013).

Each method has strengths and limitations. The method proposed by (Yao, Peng, and Tsakiris 2021) works well for sparsely permuted data or low-dimensional subspaces but struggles with missing data, whereas the method proposed by (Tang et al. 2021) handles such data but assumes permuted clusters of outliers, a condition that may not always hold. Additionally, the method of (Tang et al. 2021) is sensitive to initial conditions and assumes a low-rank subspace, with a time complexity of $\mathcal{O}(n^2)$, limiting its scalability in high-dimensional scenarios. Importantly, both methods focus on a single ground-truth subspace, overlooking the more complex scenario involving multi-subspace data (widely existing in many areas such as computer vision and signal processing), which we aim to address. While the robust kernel PCA proposed by (Fan and Chow 2019) removes sparse noise from high-rank matrices, it does not effectively handle permuted data.

1.3 Contributions of This Work

We generalize the traditional single-subspace permuted matrix recovery problem into multi-subspace scenarios and propose a four-step pipeline for recovering a multi-subspace matrix corrupted by permutation. Within the pipeline, we introduce an efficient method for the outlier classification step and provide theoretical guarantees to support the algorithm and practical applications.

*Corresponding author.

2 Methodology

2.1 Problem Formulation

Let $\mathbf{G} \in \mathbb{R}^{M\times N}$ be a clean data matrix of which the columns are randomly drawn from a union of L r-dimensional subspaces $\{\mathcal{S}_k\}_{k=1}^{L}$, where $1 \le r < M$. Suppose $N_{\mathbf{Y}}$ columns of $\mathbf{G}$, forming a matrix $\mathbf{Y} \in \mathbb{R}^{M\times N_{\mathbf{Y}}}$, are corrupted by permutations, that is, for $i \in [N_{\mathbf{Y}}]$,

$$\tilde{\mathbf{y}}_i = \mathbf{P}_i \mathbf{y}_i \tag{1}$$

where $\mathbf{P}_i \in \mathcal{P}_M$ is a partial permutation matrix. An illustrative example for $\mathbf{P}_i$ when $M = 3$ is $[1\ 0\ 0; 0\ 0\ 1; 0\ 1\ 0]$. The rest $N_{\mathbf{X}}$ columns of $\mathbf{G}$, forming a matrix $\mathbf{X} \in \mathbb{R}^{M\times N_{\mathbf{X}}}$, remain unchanged, where $N_{\mathbf{X}} + N_{\mathbf{Y}} = N$. Then the corrupted data matrix is denoted as $\widetilde{\mathbf{G}}$, consisting the columns of $\mathbf{X}$ and $\widetilde{\mathbf{Y}}$, where $\widetilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \ldots, \tilde{\mathbf{y}}_{N_{\mathbf{Y}}}]$. We call the columns of $\widetilde{\mathbf{Y}}$ *outliers* for convenience. Our goal is to recover $\mathbf{G}$ from $\widetilde{\mathbf{G}}$. This task is more challenging if L, r, or $\frac{N_{\mathbf{Y}}}{N}$ is larger. It has numerous real applications, e.g.:

- **Data Cleaning:** In many real scenarios such as health care (Talebi et al. 2020), the samples in a dataset may be drawn from a union of subspaces (corresponding to different groups or clusters) and the attribute names of many samples may be missing or incorrect due to recording mistakes or technical errors. It is important to identify these samples and recover the true orders of the attributes, such that the performance of downstream tasks is reliable.
- **Multi Dataset De-anonymization:** Data permutation methods are pivotal in data privacy and anonymization, particularly with multisubspace data (Byun et al. 2006) (Ji, Mittal, and Beyah 2016). De-anonymization challenges escalate with the presence of multiple subspaces or latent structures within datasets, which can be exploited for re-identification. This risk intensifies in multi-subspace contexts due to potential overlapping information. Consequently, adapting de-anonymization techniques to address multi-subspace scenarios is essential to mitigate privacy breaches effectively.

2.2 Proposed Method

Using a single technique such as a denoising algorithm (e.g. robust PCA) to directly recover $\mathbf{G}$ from $\widetilde{\mathbf{G}}$ is often infeasible because several important and commonly-used assumptions such as I.I.D. and low-rankness do not hold in the problem. For example, when there is no overlap between the subspaces, $\mathbf{G}$ is full-rank if $Lr \ge M$. Therefore, we propose a pipeline consisting of four stages to address the challenge. The four stages are outlier detection, subspace clustering and estimation, outlier classification, and matrix recovery, respectively. Our method, termed **P**ermuted **M**ulti-**S**ubspace **D**ata **R**ecovery (PMSDR), is summarized in Algorithm 1. In the following context, we elaborate on the four stages.

Step 1: Outlier Detection from $\widetilde{\mathbf{G}}$ Outlier detection (Hodge and Austin 2004) is a critical step in numerous data analysis tasks. Over the past few decades, a variety of methods have been developed to address this problem, ranging from traditional statistical approaches to more advanced machine learning techniques (Boukerche, Zheng, and Alfandi 2020). While these methods can be effective in certain scenarios, they may face challenges when dealing with high-dimensional or complex data. One may consider robust PCA (Candès et al. 2011; Fan et al. 2019) but it requires the low-rank assumption, which does not hold in our problem.

Algorithm 1: Permuted Multi-Subspace Data Recovery Pipeline (**PMSDR**)

Input:

- Observed matrix $\widetilde{\mathbf{G}}$ consisting the columns of $\mathbf{X}$ and $\widetilde{\mathbf{Y}}$
- Recovery rank r
- Subspace number L

Output:

- Recovered data matrix $\widehat{\mathbf{G}}$

1: **Step 1: Outlier Detection.** Seperate columns of $\widetilde{\mathbf{G}}$ into $\widehat{\mathbf{X}}$ and $\widehat{\widetilde{\mathbf{Y}}}$ using the l_1-norm of sparse self-representation coefficients (You, Robinson, and Vidal 2017).
2: **Step 2: Subspace Reconstruction.** Cluster columns of $\widehat{\mathbf{X}}$ into L groups to obtain matrices $\{\widehat{\mathbf{X}}_1, \ldots, \widehat{\mathbf{X}}_L\}$ and estimate an r-dim basis $\widehat{\mathbf{U}}_k$ for each $\widehat{\mathbf{X}}_k$ using SVD.
3: **Step 3: Outlier Classification.** Associate each outlier $\tilde{\mathbf{y}}_j$ with its respective subspace $\mathcal{S}_{k_j}$ using Algorithm 2.
4: **Step 4: Matrix Recovery.** Recover each $\tilde{\mathbf{y}}_j$ w.r.t. its corresponding subspace $\mathcal{S}_{k_j}$ using unsupervised sensing techniques such as UPCA (our default approach) (Yao, Peng, and Tsakiris 2021).
5: **Return**: $\widehat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_{N_{\widehat{\mathbf{Y}}}}]$ and $\widehat{\mathbf{G}} = [\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}]$

To identify the outliers in $\widetilde{\mathbf{G}}$, we propose to use a method called Provable Self-Representation Matrix (PSRM) given by (You, Robinson, and Vidal 2017). We construct a self-representation matrix $\mathbf{R} = (r_{ij}) \in \mathbb{R}^{n\times n}$ by solving the following elastic net problem

$$\min_{\mathbf{r}_j} \lambda\|\mathbf{r}_j\|_1 + \tfrac{1-\lambda}{2}\|\mathbf{r}_j\|_2^2 + \tfrac{\gamma}{2}\|\tilde{\mathbf{g}}_j - \widetilde{\mathbf{G}}\mathbf{r}_j\|_2^2 \quad \text{s.t. } r_{jj} = 0,$$

and then build the transition matrix $\mathbf{P} = (p_{ij}) \in \mathbb{R}^{M\times M}$ by

$$p_{ij} = |r_{ji}| / \|\mathbf{r}_i\|_1 \quad \text{for all}\{i, j\} \subseteq [M], \tag{2}$$

thus forming a stochastic process. By initializing with an initial discrete union distribution, the probability mass will, over successive steps, provably concentrate on inliers under certain assumptions. This probabilistic behavior allows us to identify and separate outliers from inliers. We denote the estimated inliers and outliers in $\widetilde{\mathbf{G}}$ as $\widehat{\mathbf{X}}$ and $\widehat{\widetilde{\mathbf{Y}}}$, respectively.

Step 2: Subspace Reconstruction Given $\widehat{\mathbf{X}}$, we need to cluster its columns into L groups to obtain matrices $\{\widehat{\mathbf{X}}_1, \ldots, \widehat{\mathbf{X}}_L\}$ corresponding to different subspaces. There is a large literature on the issue of subspace clustering in recent decades (Elhamifar and Vidal 2013; Liu et al. 2012; Fan 2021; Cai et al. 2022). Here we simply use SSC (Elhamifar and Vidal 2013) as the subspace clustering method. One can

Algorithm 2: Outlier Classification

Input: Estimated bases for all subspaces $\widehat{\mathbf{U}}_1, \ldots, \widehat{\mathbf{U}}_L \in \mathbb{R}^{M\times r}$; One outlier sample $\tilde{\mathbf{y}} \in \mathbb{R}^M$
Initialized Parameters: Retain ratio γ; Maximum iterations *max_iter*
Output: Corresponding subspace label $t \in [L]$

Initialize elimination numbers $m = [m_1, m_2, \ldots, m_{iter}]$, where each $m_i \in \mathbb{Z}^+$ is in descending order, *iter* $\leq$ *max_iter*, and $\sum_{i=1}^{iter} m_i = \lfloor (n-r) * (1-\gamma) \rfloor$.
for $k = 1$ to L **do**
  Initialize $\boldsymbol{\nu}^{(0)} = \tilde{\mathbf{y}}$ and $\mathbf{B}^{(0)} = \widehat{\mathbf{U}}_k$.
  for $i = 1$ to *iter* **do**
    $[\hat{j}_1, \hat{j}_2, \ldots, \hat{j}_{m_i}] = \operatorname{argmax}_{j_1, j_2, \ldots, j_{m_i}} \left| \boldsymbol{\nu}^{(i-1)} - \mathbf{B}^{(i-1)} {\mathbf{B}^{(i-1)}}^{\dagger} \boldsymbol{\nu}^{(i-1)} \right|$
    Remove the $[\hat{j}_1, \hat{j}_2, \ldots, \hat{j}_{m_i}]$-th entries from $\boldsymbol{\nu}^{(i-1)}$ to get $\boldsymbol{\nu}^{(i)}$.
    Remove the $[\hat{j}_1, \hat{j}_2, \ldots, \hat{j}_{m_i}]$-th rows from $\mathbf{B}^{(i-1)}$ to get $\mathbf{B}^{(i)}$ and refined subspace $\mathcal{S}_k^{(i)}$.
  end for
  Calculate the subspace distance $d_k = 1 - \cos\left(\boldsymbol{\nu}^{(iter)}, \mathcal{S}_k^{(iter)}\right)$
end for
Determine the subspace label $t = \operatorname{argmin}_k d_k$

utilize any other method as an alternative when needed. It's also worth noting that like most subspace clustering methods, the more samples there are, the better the performance is, which indicates a future direction to enhance the performance of subspace clustering by other tricky techniques.

For subspace estimation, we utilize the Singular Value Decomposition (SVD) to compute basis vectors $\widehat{\mathbf{U}}_k$ for each subspace $\mathcal{S}_k$, i.e., $\mathbf{X}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^\top$, where we define $\widehat{\mathbf{U}}_k$ as the first r columns of $\mathbf{U}_k$. These basis vectors are arranged in descending order based on their corresponding singular values. Therefore, the process involves selecting the top r eigenvectors to form a basis for $\mathcal{S}$. Alternatives like DPCP (Zhu et al. 2019) can also be used.

Step 3: Outlier Classification To recover $\mathbf{Y}$ from $\widehat{\widetilde{\mathbf{Y}}}$ using $\{\widehat{\mathbf{U}}_k\}_{k=1}^L$, we need to find the corresponding $\widehat{\mathbf{U}}_k$ or subspace for each column of $\widehat{\widetilde{\mathbf{Y}}}$ first. This is essentially a classification task but the elements of each column in $\widehat{\widetilde{\mathbf{Y}}}$ are partially permuted, which leads to a considerable challenge.

Inspired by the UPCA method proposed by (Yao, Peng, and Tsakiris 2021), we propose an efficient algorithm to resolve the challenge, shown in Algorithm 2. It iteratively eliminates unimportant entries to identify the true correspondence between an outlier and its subspace. Given an outlier

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y}^{(\mathbf{1})} \\ \tilde{\mathbf{y}}^{(\mathbf{2})} \end{bmatrix} \tag{3}$$

with $\tilde{\mathbf{y}}^{(\mathbf{2})}$ being fully permuted, and a basis of one subspace

$$\widehat{\mathbf{U}}_k = \begin{bmatrix} \widehat{\mathbf{U}}_k^{(1)} \\ \widehat{\mathbf{U}}_k^{(2)} \end{bmatrix} \quad (k = 1, \ldots, L), \tag{4}$$

the core idea is to eliminate entries in $\tilde{\mathbf{y}}^{(\mathbf{2})}$ while retaining $\mathbf{y}^{(\mathbf{1})}$ as completely as possible. Then, we compare the cosine distance between the retained vector $\mathbf{y}^{(\mathbf{1})}$ and each subspace $\mathcal{S}_k$ using the formula:

$$d_k = 1 - \cos\left(\mathbf{y}^{(\mathbf{1})}, \widehat{\mathbf{U}}_k^{(1)} \left(\widehat{\mathbf{U}}_k^{(1)}\right)^{\dagger} \mathbf{y}^{(\mathbf{1})}\right), \tag{5}$$

and select the subspace with the minimum distance as the estimated subspace class.

Specifically, the algorithm first initializes the total number of steps *iter* and the number of eliminated entries m_i in i-th step, which significantly enhances the efficiency. For each subspace $\mathcal{S}_k$, the algorithm initializes the remaining vector $\boldsymbol{\nu}^{(0)} = \tilde{\mathbf{y}}$ and the basis matrix $\mathbf{B}^{(0)} = \widehat{\mathbf{U}}_k$. During each iteration, the algorithm performs least squares regression:

$$\hat{\boldsymbol{\nu}}^{(i-1)} = \mathbf{B}^{(i-1)} \left(\mathbf{B}^{(i-1)}\right)^{\dagger} \boldsymbol{\nu}^{(i-1)} \tag{6}$$

and removes the largest m_i entries in the residual

$$\left| \boldsymbol{\nu}^{(i-1)} - \hat{\boldsymbol{\nu}}^{(i-1)} \right| \tag{7}$$

from both $\boldsymbol{\nu}$ and $\mathbf{B}$. After completing the iterations, the subspace distance is calculated to determine the subspace label t that minimizes the cosine distance between the remaining vector $\boldsymbol{\nu}^{(\text{iter})}$ and the subspace $\mathcal{S}_k^{(\text{iter})}$.

Intuitively, the outlier classification method is easy to understand. The elimination of the largest residuals in each iteration ensures that the remaining data points better represent the underlying subspace structure. By iteratively refining the basis matrix $\mathbf{B}$ and the residual vector $\boldsymbol{\nu} - \hat{\boldsymbol{\nu}}$, the algorithm effectively isolates the outlier and aligns it with the correct subspace. From a theoretical perspective, under mild assumptions, we provide approximate guarantees for the effectiveness of this method, with experimental analysis supported, which will be elaborated in the Appendix.

Step 4: Matrix Recovery It is necessary to provide an outline of *unlabeled sensing* for completeness. In a nutshell, unlabeled sensing methods (Yao, Peng, and Tsakiris 2021; Slawski and Ben-David 2019) are proposed for solving linear equation systems with unordered measurements:

$$\mathbf{y} = \boldsymbol{\pi}\mathbf{U}\mathbf{x} \tag{8}$$

where $\boldsymbol{\pi}$ is an unknown permutation matrix, with the knowledge of $\mathbf{y}$ and $\mathbf{U}$ (Unnikrishnan, Haghighatshoar, and Vetterli 2015). Different unlabeled sensing methods are brought into practice according to the rank of the basis $\mathbf{U}$ and the type of shuffling (partially shuffled or fully shuffled).

During the matrix recovery step (step 4 in Algorithm 1), we employ an unlabeled sensing method to iteratively recover each outlier $\tilde{\mathbf{y}}_t$ with its corresponding basis $\widehat{\mathbf{U}}_t$. Alternatively, matrix recovery methods like robust PCA (Candès et al. 2011), LRR (Liu et al. 2012), and RKPCA (Fan and Chow 2019) could also be utilized to recover $[\widehat{\mathbf{X}}_t, \widehat{\mathbf{Y}}_t]$ associated with subspace $\mathcal{S}_t$.

3 Theory for Outlier Classification

In this section, we provide a theoretical guarantee for Algorithm 2. To begin with, we have the following assumptions.

Assumption 1. *The variables $(\tilde{\mathbf{y}} - \hat{\tilde{\mathbf{y}}})_1, \ldots, (\tilde{\mathbf{y}} - \hat{\tilde{\mathbf{y}}})_{M_1}$ are independent and identically distributed (i.i.d.) following a distribution denoted by ξ. Similarly, the variables $(\tilde{\mathbf{y}} - \hat{\tilde{\mathbf{y}}})_{M_1+1}, \ldots, (\tilde{\mathbf{y}} - \hat{\tilde{\mathbf{y}}})_M$ are i.i.d. following a distribution denoted by η. Both ξ and η belong to the same bell-shaped distribution cluster, with a mean value $\mu_\xi = \mu_\eta \triangleq \mu = 0$, differing only in their variances $\sigma_\xi^2 \neq \sigma_\eta^2$, which means their cumulative distribution functions satisfy:*

$$F_\xi(\sigma_\xi x) = F_\eta(\sigma_\eta x) \triangleq F(x), \tag{9}$$

where $F(x)$ is the cdf of their normalized distribution with variance $\int_{\mathbb{R}} x^2 \, dF(x) = 1$.

Assumption 2. *For the sake of brevity, we assume that the bell-shaped distribution in Assumption 1 is given by $F(x) = \Phi(x)$, where $\Phi(x)$ denotes the cdf of the standard Gaussian distribution.*

We defer the detailed discussion on the assumptions to Appendix. Now we present the following theorem:

Theorem 1. *Under Assumptions 1 and 2, and without loss of generality, let $\mathcal{A} \triangleq \{i \in \mathbb{Z}^+ : 1 \leq i \leq M_1\}$ represent the unshuffled indices and $\mathcal{O} \triangleq \{i \in \mathbb{Z}^+ : M_1 + 1 \leq i \leq M\}$ represent the shuffled indices of $\mathbf{y}$. Thus $M_1 = \#(\mathcal{A})$, $M_2 = \#(\mathcal{O}) = M - M_1$. Define*

$$\hat{j} = \operatorname{argmax}_j \left| \left(\mathbf{y} - \hat{\tilde{\mathbf{y}}} \right)_j \right|. \tag{10}$$

Then, approximately,

$$\Pr\left(\hat{j} \in \mathcal{O}\right) \approx 2\Phi\left(\frac{\sigma_\eta \rho(M_2) - \sigma_\xi \rho(M_1)}{\sqrt{\sigma_\eta^2 \psi^2(M_2) + \sigma_\xi^2 \psi^2(M_1)}} \right) - 1, \tag{11}$$

where

$$\rho(m) = F^{-1}(1 - \frac{1}{m}) = \Phi^{-1}(1 - \frac{1}{m}), \tag{12}$$

$$\psi(m) = \frac{1}{m \cdot f(\rho(m))} = \frac{1}{m \cdot \phi(\rho(m))}, \tag{13}$$

with f (or ϕ) being the pdf corresponding to F (or Φ) and approximated estimation of $\sigma_\xi^2, \sigma_\eta^2$ as follows:

$$\begin{cases} \Pr\left(\sigma_\xi^2 < C\left(\frac{r(M-r)M_2}{M^2(M-1)(M+2)}\right)\right) > 1 - \delta \\ \mathbb{E}(\sigma_\xi^2) \leq \frac{M_2 r(M-r)}{M^2(M-1)(M+2)} + \frac{\sqrt{6} M_2 r^{1/2}(M-r)^{3/2}}{M^2(M-1)(M+2)^{3/2}} + \frac{M_2^2}{M^3} \\ \mathbb{E}(\sigma_\eta^2) \geq \frac{2}{M} - \frac{2[M_2 M + M - 4](M-r)}{M^2(M-1)(M+2)} + \frac{\Gamma_r\left(\frac{r}{2}+\frac{3}{r}\right)\Gamma_r\left(\frac{M}{2}\right)}{M_2 \Gamma_r\left(\frac{M}{2}+\frac{3}{r}\right)\Gamma_r\left(\frac{r}{2}\right)} \\ \mathbb{E}(\sigma_\eta^2) \approx \frac{(2M - M_2)}{M^2}\left[\frac{(M-r)(M-r+2)}{M(M+2)} + \frac{(M_2-1)r(M-r)}{M(M-1)(M+2)}\right] \end{cases} \tag{14}$$

where $C \leq 2 + 6(1+\sqrt{2})\sqrt{\frac{M-r}{rM_2M}}$, $\delta \approx 0.0054$, and multivariate gamma function

$$\Gamma_r(x) = \pi^{\frac{r(r-1)}{4}} \prod_{i=1}^{r} \Gamma\left(x - \frac{i-1}{2}\right).$$

There are some calculation issues for σ_ξ^2 and σ_η^2, which will be detailed in Appendix. Anyway, Theorem 1 ensures that Algorithm 2 successfully recovers the subspace when initialized with the ground truth subspace basis. Specifically, the shuffled ratio $\frac{M_2}{M}$ in the retaining vector $\boldsymbol{\nu}^{(i)}$ decreases rapidly as the iteration index i increases. Consequently, $\boldsymbol{\nu}^{(i)}$ quickly aligns with the ground-truth retaining subspace $\mathcal{S}^{(i)}$ in terms of cosine distance, eventually approaching zero. Intuitively, this can be understood as the following: with the ratio of retained entries being no more than γ (as defined in Algorithm 2), we can confidently assert that most of the shuffled entries have been removed. As a result, the retaining vector $\boldsymbol{\nu}^{(iter)}$ closely approximates the ground-truth retaining subspace $\mathcal{S}_{gt}^{(iter)}$.

Conversely, if the process starts with an entirely incorrect subspace, it is analogous to a situation where the data points have been completely shuffled, as discussed in the Appendix. In such a case, the entries are eliminated in a seemingly random fashion, making it impossible to distinguish between shuffled and unshuffled entries. Consequently, Algorithm 2 would fail, as corroborated by our theoretical analysis. In this scenario, the retaining vector $\boldsymbol{\nu}^{(i)}$ will not converge towards the incorrect subspace $\mathcal{S}_{wrong}^{(i)}$ at the same rate as when the correct subspace is used. This is because the retaining vector $\boldsymbol{\nu}^{(iter)}$ bears little correlation with $\mathcal{S}_{wrong}^{(iter)}$, given that the retained entries belong to an unrelated subspace. Thus, by selecting an appropriate stopping criterion γ in Algorithm 2, we can effectively differentiate the ground truth subspace label by comparing the cosine distances between each retaining vector $\boldsymbol{\nu}_k^{(i)}$ and its corresponding subspace $\mathcal{S}_k^{(i)}$, for $k = 1, \ldots, L$.

4 Experimental Evaluation

Before presenting the results of experiments conducted on synthetic and real-world datasets, it is important to clarify the evaluation metrics used to demonstrate the effectiveness of our approach.

Outlier Classification Error involves two metrics, $\mathbf{CE_{gt}}$ and $\mathbf{CE_{recon}}$, that measure outlier classification ac-

(L, p)	Permutation Error Ratio(%)
(2, 2)	**MRUC-S**: 0.0 ± 0.0 ([0, 0]) **MRUC**: 12.0 ± 15.5 ([0, 30])
(3, 2)	**MRUC-S**: 0.0 ± 0.0 ([0, 0]) **MRUC**: 30.0 ± 0.0 ([30, 30])
(3, 3)	**MRUC-S**: 13.3 ± 16.6 ([0, 33.3]) **MRUC**: 55.0 ± 1.4 ([53.3, 58.3])
(3, 5)	**MRUC-S**: 17.0 ± 25.9 ([0, 65]) **MRUC**: 71.5 ± 13.7 ([56.7, 91.7])
(5, 5)	**MRUC-S**: 23.0 ± 17.9 ([0, 50]) **MRUC**: 73.2 ± 0.6 ([73, 75])

Table 1: Performance comparison of MRUC and MRUC-S. Values represent the permutation error ratio (%), which is the average normalized Hamming distance between predicted and true permutation matrices. Metrics are reported as mean ± std ([min, max]) over multiple random initializations.

curacy. $\mathbf{CE_{gt}}$ is calculated using the ground truth subspaces, $\mathbf{U}_1, \ldots, \mathbf{U}_L$, while $\mathbf{CE_{recon}}$ uses reconstructed bases $\widehat{\mathbf{U}}_1, \ldots, \widehat{\mathbf{U}}_L$. Then:

$$\mathbf{CE_{gt}} = \frac{\#(\text{Misclassified Outliers})}{\#(\text{Outliers})}$$

$$\mathbf{CE_{recon}} = \frac{\#(\text{Misclassified Detected Outliers})}{\#(\text{Detected Outliers})}$$

Matrix Recovery Error is assessed by $\mathbf{RE_{gt}}$ and $\mathbf{RE_{recon}}$, which measure the accuracy of recovering outlier columns using normalized Frobenius norms. Specifically:

$$\mathbf{RE_{gt}} = \frac{\|\operatorname{Proj}_{\mathcal{S}}(\widehat{\mathbf{Y}}) - \mathbf{Y}\|_F}{\|\mathbf{Y}\|_F}$$

$$\mathbf{RE_{recon}} = \frac{\|\operatorname{Proj}_{\mathcal{S}}(\widehat{\mathbf{Y}}_d) - \mathbf{Y}_d\|_F}{\|\mathbf{Y}_d\|_F}$$

where $\widehat{\mathbf{Y}}$ and $\widehat{\mathbf{Y}}_d$ are recovered outliers using ground truth and detected bases, respectively, and $\mathbf{Y}$ and $\mathbf{Y}_d$ are the corresponding ground truth outliers. $\operatorname{Proj}_{\mathcal{S}}(\cdot)$ denotes projection onto the subspace.

Auxiliary Metrics include **UOratio** and **SCerr**:

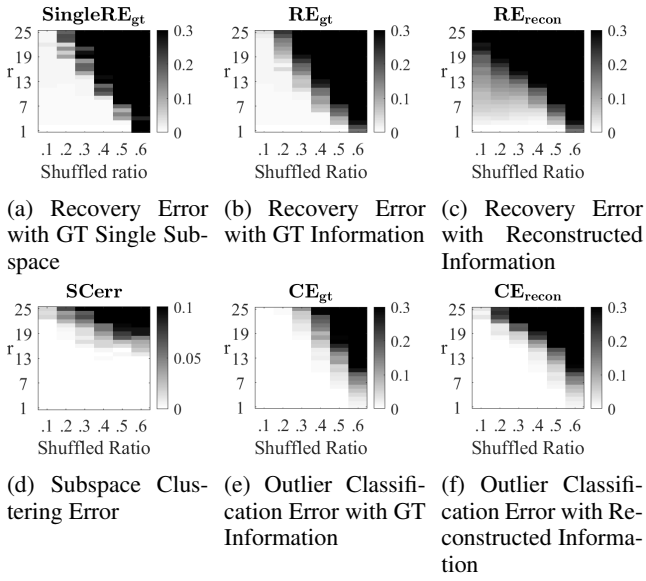
$$\mathbf{UOratio} = \frac{\#(\text{Undetected Outliers})}{\#(\text{Outliers})}$$

$$\mathbf{SCerr} = \frac{\#(\text{Misclassified Detected Inliers})}{\#(\text{Detected Inliers})}$$

UOratio evaluates the undetected ratio of outlier detection, while **SCerr** assesses subspace clustering error, both influencing $\mathbf{CE_{recon}}$ and $\mathbf{RE_{recon}}$.

4.1 Experiment on Synthetic Data

We conduct three experiments: (1) analyzing Algorithm 1, (2) comparing it with RPCA (Candès et al. 2011), RKPCA (Fan and Chow 2019), and SSC (Elhamifar and Vidal 2013), and (3) evaluating the impact of MRUC (Tang et al. 2021) on permutation matrix recovery when Algorithm 1 is augmented for multiple subspaces.



(a) Recovery Error with GT Single Subspace (b) Recovery Error with GT Information (c) Recovery Error with Reconstructed Information

(d) Subspace Clustering Error (e) Outlier Classification Error with GT Information (f) Outlier Classification Error with Reconstructed Information

Figure 1: Performance of PMSDR (Algorithm 1) on Synthetic Data. Experiments are conducted for sparse permutations with shuffled ratios up to 0.6 and subspace dimensions up to 25 with the ambient space dimension being 50. In multi-subspace cases, the number of subspaces is $2, 3, 5, 8$, or 10, and the median error is plotted.

In the first experiment, the ambient dimension M is 50, each subspace has 120 samples, and the outlier proportion is 60%. The number of subspaces is $2, 3, 5, 8$, or 10. The subspace rank varies from 1 to 25, and the shuffled ratio from 0.1 to 0.6, with a noise level of 40 dB. The median error across all settings is recorded. Figure 1 compares the estimation error of our method with single subspace results. Figure 1(a) shows UPCA (Yao, Peng, and Tsakiris 2021) on a single subspace as a baseline, while (b) and (c) show multi-subspace results, demonstrating minimal performance loss even with reconstructed information. Figures (e) and (f) highlight the robust outlier classification.

In the second experiment, we compare vanilla methods with their PMSDR-augmented versions, with the number of subspaces fixed at 5. Figure 2 shows a significant enhancement in RPCA and SSC, and a slight improvement for RKPCA when augmented. The third experiment compares PMSDR-augmented MRUC-S with MRUC using Hamming distance as the evaluation metric. The experimental setup includes $M = 20$, $r = 2$, and a shuffled ratio of 0.5. Table 1 demonstrates the superior performance of PMSDR, highlighting the robustness of Algorithm 1, even under varying subspace configurations.

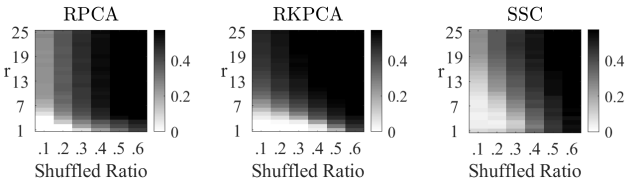
In summary, our method effectively bridges multi- and single-subspace cases in low-rank scenarios, generalizing single-subspace recovery to multi-subspace contexts.

4.2 Experiment on Face Images

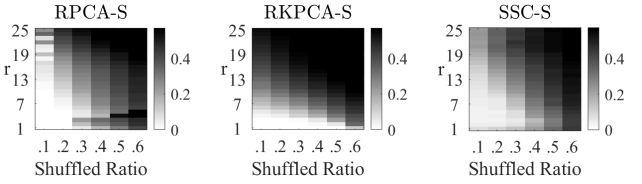
We applied our algorithm to the Extended Yale B dataset (Georghiades, Belhumeur, and Kriegman 2001), which in-

Mean (Median)	$\mathbf{CE}_{\text{gt}}$	$\mathbf{CE}_{\text{recon}}$	UOratio	SCerr
2 subjects	0.0000 (0.0000)	0.0000 (0.0000)	0.1042 (0.1095)	0.0137 (0.0053)
3 subjects	0.0000 (0.0000)	0.0110 (0.0105)	0.0488 (0.0525)	0.0106 (0.0072)
5 subjects	0.0168 (0.0190)	0.0343 (0.0325)	0.0333 (0.0250)	0.0493 (0.0304)
8 subjects	0.0245 (0.0230)	0.0595 (0.0630)	0.0157 (0.0120)	0.0959 (0.1007)
10 subjects	0.0250 (0.0250)	0.1002 (0.0845)	0.0107 (0.0130)	0.1336 (0.1563)
12 subjects	0.0268 (0.0285)	0.1025 (0.1030)	0.0093 (0.0100)	0.1409 (0.1892)

Table 2: Performance of Experiment on Extended YaleB Dataset



(a) Recovery Error for RPCA (b) Recovery Error for RKPCA (c) Recovery Error for SSC



(d) Recovery Error for RPCA-S (e) Recovery Error for RKPCA-S (f) Recovery Error for SSC-S

Figure 2: Synthetic experiments on RPCA, RKPCA, SSC, and their PMSDR-augmented versions. All experiments are conducted for sparse permutations with shuffled ratios no greater than 0.6 and subspace dimensions no greater than 50% of the ambient space dimension, which is 50. The number of subspaces is fixed to 5.

cludes 38 subjects, each with 64 downsampled face images of size 48×42 ($M = 2016$).

In the first experiment, we selected 10 subjects and corrupted 19 images per group by shuffling 40% of the pixels. We compared our PMSDR method with RPCA (Candès et al. 2011), SSC (Elhamifar and Vidal 2013), and RKPCA (Fan and Chow 2019), as well as their PMSDR-augmented counterparts (RPCA-S, SSC-S, RKPCA-S). The results, depicted in Figure 6, demonstrate that PMSDR substantially improves matrix recovery and outlier correction. A subset of these results is presented in Figure 3, with the complete set available in Appendix B.1.

In the second experiment, we varied the number of subspaces L from 2 to 12 and repeated the corruption process. The results, summarized in Table 2, indicate a strong performance, particularly when the ground truth is known. Specifically, $\mathbf{CE}_{\mathbf{gt}}$ is lower than $\mathbf{CE}_{\mathbf{recon}}$, due to increased subspace clustering errors **SCerr**. However, these findings highlight the robustness and adaptability of our algorithm.

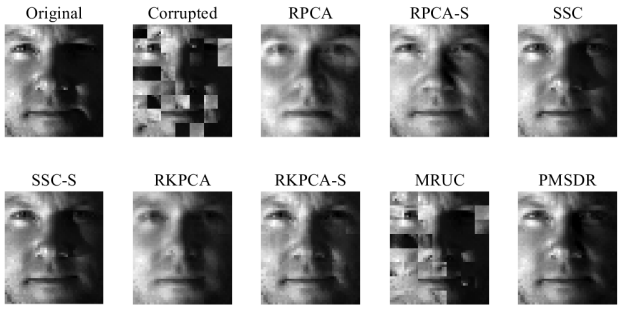


Figure 3: Experimental results showing a subset of the image recovery experiments. The complete set of results is in Appendix B.1.

Subspaces	Metric	Mean	Median
2 subspaces	$\mathbf{CE}_{\text{gt}}$	0.011	0.000
	$\mathbf{CE}_{\text{recon}}$	0.040	0.007
	$\mathbf{RE}_{\text{gt}}$	0.017	0.005
	$\mathbf{RE}_{\text{recon}}$	0.021	0.013
	SCerr	0.027	0.000
3 subspaces	$\mathbf{CE}_{\text{gt}}$	0.018	0.008
	$\mathbf{CE}_{\text{recon}}$	0.093	0.023
	$\mathbf{RE}_{\text{gt}}$	0.014	0.007
	$\mathbf{RE}_{\text{recon}}$	0.021	0.012
	SCerr	0.043	0.004

Table 3: PMSDR Performance on Hopkins-155

4.3 Experiment on Motion Segmentations

We evaluated our algorithm on the Hopkins-155 database, which includes 117 sequences with 2 subspaces, 35 with 3 subspaces and 1 with 5 subspaces. Each sequence lies in a 4-dimensional subspace (Boult and Brown 1991; Tomasi and Kanade 1992). The shuffled and outlier ratios are both 0.4, with a fixed subspace dimension of 4. Data are mapped from 3D to 2D for better representation, and concatenated over frames. After preprocessing, the dimension of the ambient space ranges from 40 to 70.

We apply our 4-stage pipeline (PMSDR) and compare its performance against RPCA, RKPCA, SSC, and MRUC, along with their PMSDR-augmented variants, denoted by appending the suffix '-S'. Additionally, we investigate the

Method	2 subspaces	3 subspaces
Median RE (Mean RE)		
PMSDR	0.013 (0.021)	0.012 (0.021)
PMSDR_{gt}	**0.005 (0.017)**	**0.007 (0.014)**
RPCA	0.046 (0.052)	0.047 (0.055)
RPCA-S	0.040 (0.045)	0.043 (0.047)
RPCA_{gt} -S	**0.033 (0.040)**	**0.030 (0.036)**
RKPCA	**0.009 (0.014)**	**0.008 (0.012)**
RKPCA-S	0.053 (0.064)	0.047 (0.058)
RKPCA_{gt} -S	0.039 (0.057)	0.036 (0.049)
SSC	0.087 (0.090)	0.100 (0.104)
SSC-S	0.091 (0.095)	0.101 (0.102)
SSC_{gt}-S	**0.079 (0.084)**	**0.089 (0.090)**
MRUC	0.066 (0.084)	0.085 (0.091)
MRUC-S	0.048 (0.074)	0.056 (0.078)
MRUC_{gt}-S	**0.048 (0.070)**	**0.056 (0.073)**

Table 4: Matrix Recovery Error for Comparison Experiments on Hopkins-155.

impact of using ground-truth information for matrix recovery. The regularization parameters in RPCA, RKPCA, and SSC are tuned accordingly. Results in Table 3 show PMSDR's robustness in matrix recovery and outlier classification, even without ground truth information. Table 4 highlights further improvements when combining these methods with our pipeline, except for RKPCA pairs. For the experiments involving RKPCA pairs, we discovered an effective technique that greatly improves the performance of vanilla RKPCA. However, this approach showed ineffective for RKPCA-S and RKPCA_{gt}-S. The details of this technique are provided in the Appendix.

4.4 Experiment on Data Re-identification

We evaluated the proposed PMSDR pipeline alongside the RPCA, SSC, RKPCA, and MRUC methods on real-world educational and medical records, simulating a privacy protection scenario similar to (Yao, Peng, and Tsakiris 2021). The first dataset, described in (Fellegi and Sunter 1969), contains a matrix $M_{score} \in \mathbb{R}^{707\times 14}$ with scores of 707 students across 14 tests. To anonymize, the last 7 columns were randomly permuted, with shuffled ratios from 0.1 to 1. The second dataset from (Dua, Graff et al. 2017) involves a matrix $M_{tumor} \in \mathbb{R}^{357\times 30}$, representing 357 patients with 30 features. Here, 50% of the columns were permuted with similar shuffled ratios. Both matrices were normalized, and our method was applied with a subspace dimension of 3 as in (Yao, Peng, and Tsakiris 2021). We assumed prior knowledge of the outlier ratio during the detection phase (Step 1 in Algorithm 1).

Given the lack of inherent multi-subspace scenarios in these datasets, no ground truth subspace information is available. Thus, even single subspace methods can perform reasonably well. We compared our PMSDR with RPCA (λ optimized over $0.1 : 0.05 : 0.95$ and $\mu = 10\lambda$), RKPCA (λ optimized similarly), SSC (α optimized over $[5, 20, 100, 200, 500, 1000]$), and MRUC with the best initialization. UPCA (Yao, Peng, and Tsakiris 2021), a strong single-subspace recovery method, was used as a baseline.

In PMSDR, we hypothesized subspaces $L = 1, 2, 3$ despite the lack of natural subspaces. Figures 4 and 5 show that PMSDR outperforms UPCA in most cases, with slightly worse performance in low shuffled ratios due to possible undetected outliers affecting the basis of the subspace. This suggests that Algorithm 1 can uncover more hidden information, improving robustness and performance.

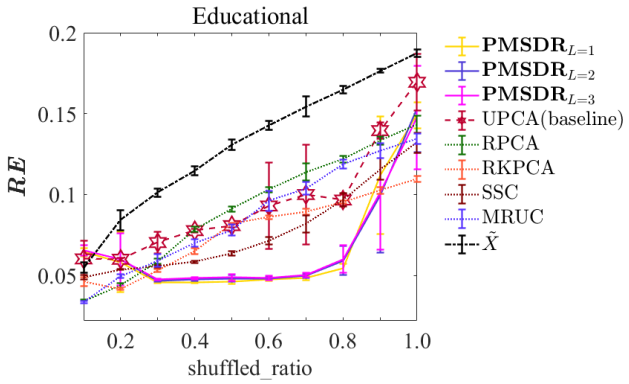


Figure 4: De-Anonymization Experiments for Educational Data Comparing Algorithm 1, UPCA (Yao, Peng, and Tsakiris 2021), RPCA, RKPCA, SSC, and MRUC. The output of Algorithm 1 is denoted as $\textbf{PMSDR}_{L\,=\,k}$, where the number of groups k is set to $\{1, 2, 3\}$.

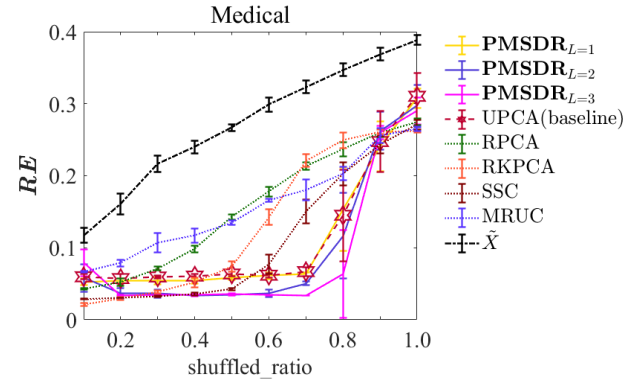


Figure 5: De-Anonymization Experiments for Medical Data Comparing Algorithm 1, UPCA (Yao, Peng, and Tsakiris 2021), RPCA, RKPCA, SSC, and MRUC. The output of Algorithm 1 is denoted as $\textbf{PMSDR}_{L\,=\,k}$, where the number of groups k is set to $\{1, 2, 3\}$.

5 Conclusion and Future Directions

Our algorithm extended UPCA (Yao, Peng, and Tsakiris 2021) by recovering corrupted data across multiple subspaces. For $L > 1$, complexity increases due to outlier matching. Instead of the robust PCA, we use outlier detection and subspace clustering to estimate bases. While our outlier classification (Step 3 in Algorithm 1, which detailed in Algorithm 2) performs well, it depends on accurate basis estimation, revealing potential areas for improvement.

The framework is flexible, integrating other methods, but currently handles only linear/affine cases. Extending it to non-linear contexts requires further research. The algorithm is also limited to partially shuffled data, a restriction future work should address. Applying permutation recovery methods like MRUC (Tang et al. 2021) as preprocessing could transform fully shuffled data into a partially shuffled state, enabling our PMSDR pipeline to function effectively.

In conclusion, our method enhances recovery across multiple subspaces, but further research is needed to improve basis estimation, handle non-linear scenarios, and overcome the partially shuffled data limitation.

Acknowledgments

This work was supported by the Shenzhen Science and Technology Program under Grant No.JCYJ20210324130208022 (Fundamental Algorithms of Natural Language Understanding for Chinese Medical Text Processing) and the Youth Program 62106211 of the National Natural Science Foundation of China.

References

Boukerche, A.; Zheng, L.; and Alfandi, O. 2020. Outlier detection: Methods, models, and classification. *ACM Computing Surveys (CSUR)*, 53(3): 1–37.

Boult, T. E.; and Brown, L. G. 1991. Factorization-based segmentation of motions. In *Proceedings of the IEEE workshop on visual motion*, 179–180. IEEE Computer Society.

Byun, J.-W.; Sohn, Y.; Bertino, E.; and Li, N. 2006. Secure anonymization for incremental datasets. In *Secure Data Management: Third VLDB Workshop, SDM 2006, Seoul, Korea, September 10-11, 2006. Proceedings 3*, 48–63. Springer.

Cai, J.; Fan, J.; Guo, W.; Wang, S.; Zhang, Y.; and Zhang, Z. 2022. Efficient Deep Embedded Subspace Clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–10.

Candès, E. J.; Li, X.; Ma, Y.; and Wright, J. 2011. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 1–37.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Domingo-Ferrer, J.; and Muralidhar, K. 2016. New directions in anonymization: permutation paradigm, verifiability by subjects and intruders, transparency to users. *Information Sciences*, 337: 11–24.

Dua, D.; Graff, C.; et al. 2017. UCI machine learning repository.

Elhamifar, E.; and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11): 2765–2781.

Fan, J. 2021. Large-Scale Subspace Clustering via k-Factorization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, 342–352. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383325.

Fan, J.; and Chow, T. W. 2018. Non-linear matrix completion. *Pattern Recognition*, 77: 378–394.

Fan, J.; and Chow, T. W. 2019. Exactly robust kernel principal component analysis. *IEEE transactions on neural networks and learning systems*, 31(3): 749–761.

Fan, J.; Ding, L.; Chen, Y.; and Udell, M. 2019. Factor group-sparse regularization for efficient low-rank matrix recovery. *Advances in neural information processing Systems*, 32.

Fan, J.; and Udell, M. 2019. Online high rank matrix completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8690–8698.

Fan, J.; Zhang, Y.; and Udell, M. 2020. Polynomial matrix completion for missing data imputation and transductive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3842–3849.

Fellegi, I. P.; and Sunter, A. B. 1969. A theory for record linkage. *Journal of the American Statistical Association*, 64(328): 1183–1210.

Georghiades, A. S.; Belhumeur, P. N.; and Kriegman, D. J. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6): 643–660.

Hodge, V.; and Austin, J. 2004. A survey of outlier detection methodologies. *Artificial intelligence review*, 22: 85–126.

Ji, S.; Mittal, P.; and Beyah, R. 2016. Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey. *IEEE Communications Surveys & Tutorials*, 19(2): 1305–1326.

Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; and Ma, Y. 2012. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 171–184.

Muralidhar, K. 2017. Record re-identification of swapped numerical microdata. *Journal of Information Privacy and Security*, 13(1): 34–45.

Slawski, M.; and Ben-David, E. 2019. Linear regression with sparsely permuted data.

Talebi, Y.; Feng, H.; Huang, Y.; and Maroufy, V. 2020. EHR data cleaning. In *Statistics and Machine Learning Methods for EHR Data*, 79–109. Chapman and Hall/CRC.

Tang, Z.; Chang, T.-H.; Ye, X.; and Zha, H. 2021. Low-rank Matrix Recovery With Unknown Correspondence. *arXiv preprint arXiv:2110.07959*.

Tomasi, C.; and Kanade, T. 1992. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2): 137–154.

Unnikrishnan, J.; Haghighatshoar, S.; and Vetterli, M. 2015. Unlabeled sensing: Solving a linear system with unordered measurements. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 786–793. IEEE.

Yao, Y.; Peng, L.; and Tsakiris, M. 2021. Unlabeled principal component analysis. *Advances in Neural Information Processing Systems*, 34.

You, C.; Robinson, D. P.; and Vidal, R. 2017. Provable self-representation based outlier detection in a union of subspaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3395–3404.

Zhu, Z.; Ding, T.; Robinson, D.; Tsakiris, M.; and Vidal, R. 2019. A linearly convergent method for non-smooth non-convex optimization on the grassmannian with applications to robust subspace and dictionary learning. *Advances in Neural Information Processing Systems*, 32.