Fast Stagewise Sparse Factor Regression

Kun Chen   KUN.CHEN@UCONN.EDU
Department of Statistics
University of Connecticut
Storrs, CT, USA

Ruipeng Dong
International Institute of Finance, The School of Management,
University of Science and Technology of China
Hefei, Anhui, China

Wanwan Xu
Department of Biostatistics
Yale University
New Haven, CT, USA

Zemin Zheng   ZHENGZM@USTC.EDU.CN
International Institute of Finance, The School of Management,
University of Science and Technology of China
Hefei, Anhui, China

Editor: Genevera Allen

Abstract

Sparse factorization of a large matrix is fundamental in modern statistical learning. In particular, the sparse singular value decomposition has been utilized in many multivariate regression methods. The appeal of this factorization is owing to its power in discovering a highly-interpretable latent association network. However, many existing methods are either ad hoc without a general performance guarantee, or are computationally intensive. We formulate the statistical problem as a sparse factor regression and tackle it with a two-stage "deflation + stagewise learning" approach. In the first stage, we consider both sequential and parallel approaches for simplifying the task into a set of co-sparse unit-rank estimation (CURE) problems, and establish the statistical underpinnings of these commonly-adopted and yet poorly understood deflation methods. In the second stage, we innovate a contended stagewise learning technique, consisting of a sequence of simple incremental updates, to efficiently trace out the whole solution paths of CURE. Our algorithm achieves a much lower computational complexity than alternating convex search, and it enables a flexible and principled tradeoff between statistical accuracy and computational efficiency. Our work is among the first to enable stagewise learning for non-convex problems, and the idea can be applicable in many multi-convex problems. Extensive simulation studies and an application in genetics demonstrate the effectiveness and scalability of our approach.

Keywords: Biconvex, high-dimensional learning, multivariate analysis, singular value decomposition, stagewise estimation

---

Authors are listed in alphabetical order.

1. Introduction

Matrix factorization/decomposition is fundamental in many statistical learning techniques, including both unsupervised learning methods such as principal component analysis and matrix completion, and supervised learning methods such as partial least squares and reduced-rank regression. In particular, singular value decomposition (SVD) and its various generalizations and extensions have been applied or utilized for matrix/tensor data dimensionality reduction, feature extraction, anomaly detection, data visualization, among others. Such tasks are nowadays routinely encountered in various fields, and some modern applications include latent semantic analysis in natural language processing (Landauer et al., 1998), collaborative filtering in recommender systems (Adomavicius and Tuzhilin, 2005), bi-clustering analysis with high-throughput genomic data (Lee et al., 2010; Vounou et al., 2010), association network learning in expression quantitative trait loci (eQTLs) mapping analysis (Uematsu et al., 2019), efficient parameterization and stabilization in the training of deep neural networks (Sun et al., 2017), and analyzing large-scale genetic and phenotypic data from UK Biobank (Bycroft et al., 2018; Qian et al., 2022).

It is now well understood that classical SVD and its associated multivariate techniques have poor statistical properties in high dimensional problems, as the noise accumulation due to a large number of irrelevant variables and/or irrelevant latent subspace directions could disguise the signal of interest to the extent that the estimation becomes entirely distorted. To break such "curses of dimensionality", structural low-dimensional assumptions such as low-rankness and sparsity are often imposed on the matrix/tensor objects, and correspondingly regularization techniques are then called to the rescue. There is no free lunch though: matrix factorization is often expensive in terms of computational complexity and memory usage, and for large-scale problems the computational burdens emerging from regularized estimation become even more amplified. The required regularized estimation can be computationally intensive and burdensome due to various reasons: the optimization often requires iterative SVD operations, and the data-driven estimation of the regularization parameters often requires a spectrum of models with varying complexities being fitted.

To alleviate the computational burden of sparse matrix factorization, many related methods have been proposed as repeatedly fitting regularized rank-one models; see, e.g., Witten et al. (2009), Lee et al. (2010), Allen et al. (2014), and Ahn et al. (2015). In particular, Chen et al. (2012) proposed the exclusive extraction algorithm which isolates the estimation of each rank-one component by removing the signals of the other components from the response matrix based on an initial estimator of the matrix factorization. Alternatively, Mishra et al. (2017) suggested extracting unit-rank factorization one by one in a sequential fashion, each time with the previously extracted components removed from the current response matrix. Both of them adapted the alternating convex search (ACS) method to solve the unit-rank estimation problems. Despite their proven effectiveness in various applications, the properties of these deflation strategies are not well understood in a general high-dimensional setup. Moreover, for each unit-rank problem, the optimization needs to be repeated for a grid of tuning parameters for locating the optimal factorization along the paths, and even with a warm-start strategy, it can still lead to skyrocketing computational costs especially for large-scale problems.

In parallel to the aforementioned learning scheme of "regularization + optimization" in solving the unit-rank problems, there has been a revival of interest into the so-called *stagewise learning* (Efron et al., 2004; Zhao and Yu, 2013; Tibshirani, 2015; He et al., 2018). Unlike regularized estimation, a stagewise procedure builds a model from scratch, and gradually increases the model

complexity in a sequence of simple learning steps. For instance, in stagewise linear regression (Efron et al., 2004; Hastie et al., 2008), a *forward step* searches for the best predictor to explain the current residual and updates its coefficient by a small increment (in magnitude), and a *backward step* may decrease the coefficient (in magnitude) of a previously selected predictor to correct for greedy selection. This process is repeated until a model with a desirable complexity level is reached or the model becomes excessively large and ceases to be identifiable. Thus far, fast stagewise learning algorithms and frameworks have been developed mainly for convex problems, which are not applicable for non-convex problems. A comprehensive review of stagewise learning can be found in Tibshirani (2015), in which its connections and differences with various optimization and machine learning approaches such as steepest descend, boosting, and path-following algorithms were discussed.

In this paper, we formulate the statistical problem of sparse factorization of large matrices as a *sparse factor regression* and tackle it with a novel two-stage approach. In the first stage, we consider both sequential and parallel approaches for simplifying the task into a set of co-sparse unit-rank estimation (CURE) problems and establish the statistical underpinnings of these commonly adopted and yet poorly understood deflation methods. In the second stage, we innovate a contended stagewise learning technique, consisting of a sequence of simple incremental updates, to efficiently trace out the whole solution paths of CURE. Our algorithm has a much lower computational complexity than ACS and the choice of the step size enables a flexible and principled tradeoff between statistical accuracy and computational efficiency. Our work is among the first to enable stagewise learning for non-convex problems, and the idea can be applicable in many bi-convex problems including sparse factorization of a tensor. Extensive simulation studies and an application in genetics demonstrate the effectiveness and scalability of our approach.

The rest of the paper is organized as follows. In Section 2, we present a general statistical setup with a sparse SVD formulation, and provide the deflation strategies for reducing the general problem into a set of simpler CURE problems with comprehensive theoretical properties. Section 3 provides detailed derivations of the stagewise estimation algorithm for tracing out the solution paths of CURE. Algorithmic convergence and computational complexity analysis are also shown in Section 3. In Section 4, extensive simulation studies demonstrate the effectiveness and scalability of our approach. An application in genetics is presented in Section 5. Section 6 concludes with possible future work. All technical details are relegated to the Appendix.

2. A Statistical Framework for Sparse Factorization

Throughout the paper, we use the bold uppercase letter to denote the matrix and the bold lowercase letter to denote the column vector. For any matrix $\mathbf{M} = (m_{ij})$, denote by $\|\mathbf{M}\|_F = \left(\sum_{i,j} m_{ij}^2\right)^{1/2}$, $\|\mathbf{M}\|_1 = \sum_{i,j} |m_{ij}|$, and $\|\mathbf{M}\|_\infty = \max_{i,j} |m_{ij}|$ the Frobenius norm, entrywise ℓ_1-norm, and entrywise ℓ_∞-norm, respectively. We also denote by $\|\cdot\|_2$ the induced matrix norm (operator norm), which becomes the ℓ_2 norm for vectors.

2.1 Co-Sparse Factor Regression

We start with the population model for the multivariate reduced-rank regression problem (Anderson, 1951; Reinsel and Velu, 1998) to motivate our study,

$$\mathbf{y} = \mathbf{C}^{*\mathrm{T}}\mathbf{x} + \mathbf{e},$$

3

where $\mathbf{y} \in \mathbb{R}^q$ is the multivariate response vector, $\mathbf{x} \in \mathbb{R}^p$ is the multivariate predictor vector, $\mathbf{C}^* \in \mathbb{R}^{p \times q}$ is the coefficient matrix that is assumed to be of low-rank, i.e., $\text{rank}(\mathbf{C}^*) = r^*$ with $r^* \leq \min(p, q)$, and $\mathbf{e} \in \mathbb{R}^q$ is the random error vector of zero mean. With a proper parameterization, this model possesses a supervised factor analysis interpretation. To see this, consider the random design and denote $\text{cov}(\mathbf{x}) = \mathbf{\Gamma}$. Write $\mathbf{C}^* = \mathbf{U}^* \widetilde{\mathbf{V}}^{\mathrm{T}}$ for some $\mathbf{U}^* \in \mathbb{R}^{p \times r^*}$ and $\widetilde{\mathbf{V}} \in \mathbb{R}^{q \times r^*}$. Then $\mathbf{U}^{*\mathrm{T}}\mathbf{x}$ can be interpreted as a set of "supervised latent factors", as some linear combinations of the components of $\mathbf{x}$; to avoid redundancy, it is desirable to make these factors uncorrelated, which leads to the constraint that $\text{cov}(\mathbf{U}^{*\mathrm{T}}\mathbf{x}) = \mathbf{U}^{*\mathrm{T}}\mathbf{\Gamma}\mathbf{U}^* = \mathbf{I}_{r^*}$. Similar to the factor analysis, we could write $\widetilde{\mathbf{V}} = \mathbf{V}^* \mathbf{D}^*$ with $\mathbf{V}^{*\mathrm{T}}\mathbf{V}^* = \mathbf{I}$ and $\mathbf{D}^*$ a diagonal matrix to ensure parameter identifiability. This leads to a generalized SVD factorization $\mathbf{C}^* = \mathbf{U}^* \mathbf{D}^* \mathbf{V}^{*\mathrm{T}}$, where $\mathbf{U}^{*\mathrm{T}}\mathbf{\Gamma}\mathbf{U}^* = \mathbf{V}^{*\mathrm{T}}\mathbf{V}^* = \mathbf{I}_{r^*}$.

With n independent observations, let $\mathbf{Y} = [\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_q] = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times q}$ be the response matrix, and $\mathbf{X} = [\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_p] = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times p}$ be the predictor matrix. We consider the sample-version of the multivariate reduced-rank regression model,

$$\mathbf{Y} = \mathbf{X}\mathbf{C}^* + \mathbf{E}, \tag{1}$$

where $\mathbf{E} \in \mathbb{R}^{n \times q}$ is the random error matrix. Throughout the paper, we consider the fixed design case and assume that each column of $\mathbf{X}$ is normalized to have ℓ_2 norm $\sqrt{n}$. Motivated by the parameterization in the population model, we assume $\mathbf{C}^*$ admits the following factorization:

$$\mathbf{C}^* = \mathbf{U}^* \mathbf{D}^* \mathbf{V}^{*\mathrm{T}}, \qquad \text{s.t. } (\frac{1}{\sqrt{n}}\mathbf{X}\mathbf{U}^*)^{\mathrm{T}}(\frac{1}{\sqrt{n}}\mathbf{X}\mathbf{U}^*) = \mathbf{V}^{*\mathrm{T}}\mathbf{V}^* = \mathbf{I}_{r^*}, \tag{2}$$

where $\mathbf{U}^* = [\mathbf{u}_1^*, \ldots, \mathbf{u}_{r^*}^*] \in \mathbb{R}^{p \times r^*}$, $\mathbf{V}^* = [\mathbf{v}_1^*, \ldots, \mathbf{v}_{r^*}^*] \in \mathbb{R}^{q \times r^*}$, and $\mathbf{D}^* = \text{diag}\{d_k^*; k = 1, \ldots, r^*\} \in \mathbb{R}^{r^* \times r^*}$ is a diagonal matrix. Here $(1/\sqrt{n})\mathbf{X}\mathbf{u}_k^*$, $k = 1, \ldots, r^*$ can be regarded a set of orthogonal factors, each of which is constructed as a linear combination of the p predictors. Furthermore, when both $\mathbf{U}^*$ and $\mathbf{V}^*$ are assumed to be sparse, we say that $\mathbf{C}^*$ admits a (generalized) co-sparse SVD structure and call the resulting model as a *co-sparse factor regression* (Mishra et al., 2017). The interpretation is very appealing in statistical modeling: the q outcomes are related to the p features only through r^* distinct pathways, each of which may only involve small subsets of the outcomes and the features.

For convenience, we denote $\mathbf{C}_k^* = d_k^* \mathbf{u}_k^* \mathbf{v}_k^{*\mathrm{T}}$ for $k = 1, \ldots, r^*$, where $\mathbf{u}_k^*, \mathbf{v}_k^*$ are the kth column of $\mathbf{U}^*, \mathbf{V}^*$ respectively and d_k^* is the kth diagonal element of $\mathbf{D}^*$. In addition, let $\mathbf{P} = n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}$. Then we call $\mathbf{C}_k^*$ as the kth layer $\mathbf{P}$-orthogonal SVD of $\mathbf{C}^*$ (Zheng et al., 2019). We stress that the factorization in (1) is a generalized SVD of $\mathbf{C}^*$, not its regular SVD. In fact, besides the motivation from factor analysis and reduced-rank regression, this factorization can also be regarded as arising from performing the SVD of the scaled regression component $(1/\sqrt{n})\mathbf{X}\mathbf{C}^*$. That is, $(1/\sqrt{n})\mathbf{X}\mathbf{C}^* = (1/\sqrt{n})\mathbf{X}\mathbf{U}^*\mathbf{D}^*\mathbf{V}^{*\mathrm{T}}$, where $(1/\sqrt{n})\mathbf{X}\mathbf{U}^*$ and $\mathbf{V}^*$ are the left and right singular matrices of $(1/\sqrt{n})\mathbf{X}\mathbf{C}^*$, respectively. Notice that when setting $\mathbf{X}$ as the $n \times n$ identity matrix, the above general model subsumes the matrix approximation problem in the unsupervised learning scenario. Further, if $\mathbf{Y}$ is partially observed, the model connects to the matrix completion problem and the trace regression (Hastie et al., 2015).

Our main focus is then the recovery of the $\mathbf{P}$-orthogonal SVD of $\mathbf{C}^*$ under the co-sparse factor model as specified in (1) and (2). We stress that we desire the elements of the matrix factorization, e.g., the left and right singular vectors, to be entrywisely sparse, rather than only making $\mathbf{C}^*$ to be sparse. A matrix with sparse singular vectors could be sparse itself, but not vice versa in general. The many appealing features of the sparse SVD factorization have been well demonstrated in

various scientific applications (Vounou et al., 2010; Chen et al., 2014; Ma et al., 2014; Qian et al., 2022).

2.2 Unit-Rank Deflation

The simultaneous presence of the low rank and the co-sparsity structure makes the sparse SVD recovery challenging. A joint estimation of all the sparse singular vectors may necessarily involve identifiability constraints such as orthogonality (Vounou et al., 2010; Uematsu et al., 2019), which makes the optimization computationally intensive. Motivated by the power method for computing SVD (Golub and Van Loan, 1996), we take a deflation approach to tackle the problem. The main idea is to estimate the unit-rank components of $\mathbf{X}\mathbf{C}^*$ one by one, thus reducing the problem into a set of much simpler unit-rank problems.

We first present the "division" that we desire for the estimation of the co-sparse factor model in (1) and (2). That is, we aim to simplify the multi-rank factorization problem into a set of unit-rank problems of the following form:

$$\min_{\mathbf{C}\in\mathbb{R}^{p\times q}} \left\{ L(\mathbf{C};\mathcal{T}_n) + \rho(\mathbf{C};\lambda) \right\}, \quad \text{s.t. } \mathrm{rank}(\mathbf{C}) \le 1, \tag{3}$$

where $\mathcal{T}_n = \{(\mathbf{y}_i, \mathbf{x}_i); i = 1, \ldots, n\}$ denotes the observed data, $L(\mathbf{C};\mathcal{T}_n) = (2n)^{-1}\|\mathbf{Y} - \mathbf{X}\mathbf{C}\|_F^2$ is the sum of squares loss function, and $\rho(\mathbf{C};\lambda) = \lambda\|\mathbf{C}\|_1$ is the ℓ_1 penalty term with tuning parameter $\lambda \ge 0$. Intuitively, this criterion targets on the best sparse unit-rank approximation of $\mathbf{Y}$ in the column space of $\mathbf{X}$. Intriguingly, due to the unit-rank constraint, the problem can be equivalently expressed as a *co-sparse unit-rank regression* (CURE),

$$\min_{d,\mathbf{u},\mathbf{v}} \left\{ L(d\mathbf{u}\mathbf{v}^{\mathrm{T}};\mathcal{T}_n) + \lambda\|d\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_1 \right\}, \qquad \text{s.t. } d \ge 0, n^{-1/2}\|\mathbf{X}\mathbf{u}\|_2 = 1, \|\mathbf{v}\|_2 = 1. \tag{4}$$

The penalty term is multiplicative in that $\|\mathbf{C}\|_1 = \|d\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_1 = d\|\mathbf{u}\|_1\|\mathbf{v}\|_1$, conveniently producing a co-sparse factorization. This is not really surprising, because the sparsity of a unit-rank matrix directly leads to the sparsity in both its left and right singular vectors. Therefore, formulating the problem in term of the unit-rank matrix enables CURE to use a single penalty term to achieve co-sparse factorization.

We now consider two general deflation approaches, in order to reach the desired simplification in (3), or equivalently, in (4). The first approach is termed "*parallel pursuit*", which is motivated by the exclusive extraction algorithm proposed in Chen et al. (2012). This approach requires an initial estimator of the elements of the matrix factorization of $\mathbf{C}^*$; the method then isolates the estimation of each rank-one component $\mathbf{C}_k^*$ by removing from the response matrix $\mathbf{Y}$ the signals of the other components based on the initial estimator. To be specific, let $\widetilde{\mathbf{C}} = \sum_{k=1}^r \widetilde{\mathbf{C}}_k = \sum_{k=1}^r \widetilde{d}_k \widetilde{\mathbf{u}}_k \widetilde{\mathbf{v}}_k^{\mathrm{T}}$ be the initial estimator of $\mathbf{C}^*$ and its $\mathbf{P}$-orthogonal SVD as in (2). Then the parallel pursuit solves the problems

$$\widehat{\mathbf{C}}_k = \arg\min_{\mathbf{C}} \left\{ L\left(\mathbf{C} + \sum_{j\neq k}^r \widetilde{\mathbf{C}}_j; \mathcal{T}_n\right) + \rho(\mathbf{C};\lambda) \right\}, \qquad \text{s.t. } \mathrm{rank}(\mathbf{C}) \le 1, \tag{5}$$

for $k = 1, \ldots, r$, which can be implemented in parallel. See Figure 1 for an illustration.

The second deflation approach is "*sequential pursuit*", which was proposed by Mishra et al. (2017). This method extracts unit-rank factorization one by one in a sequential fashion, each time
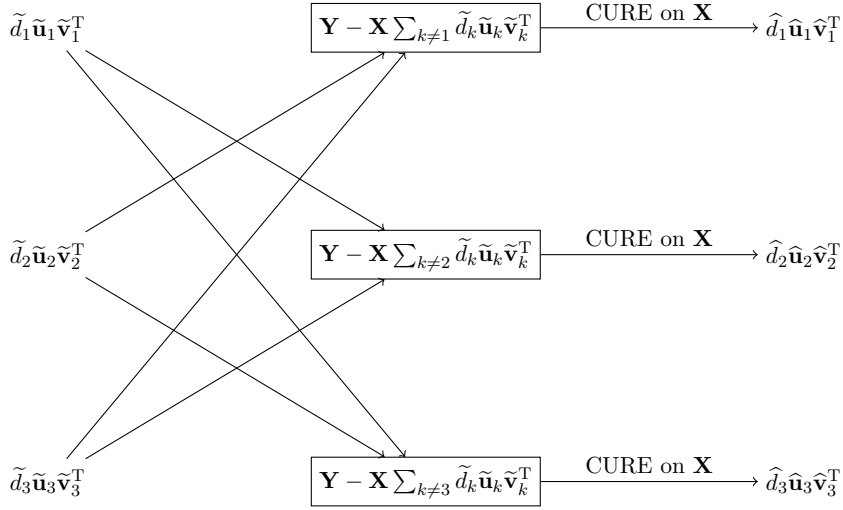
Figure 1: An illustration of the parallel pursuit.

with the previously extracted components removed from the current response matrix. Specifically, the procedure sequentially solves

$$\widehat{\mathbf{C}}_k = \arg\min_{\mathbf{C}} \left\{ L(\mathbf{C} + \sum_{j=0}^{k-1} \widehat{\mathbf{C}}_j; \mathcal{T}_n) + \rho(\mathbf{C}; \lambda) \right\}, \qquad \text{s.t. } \text{rank}(\mathbf{C}) \le 1, \tag{6}$$

for $k = 1, \dots r$, where $\widehat{\mathbf{C}}_0 = 0$, and $\widehat{\mathbf{C}}_k = \widehat{d}_k \widehat{\mathbf{u}}_k \widehat{\mathbf{v}}_k$ is the selected unit-rank solution (through tuning) in the kth step. See Figure 2 for an illustration.
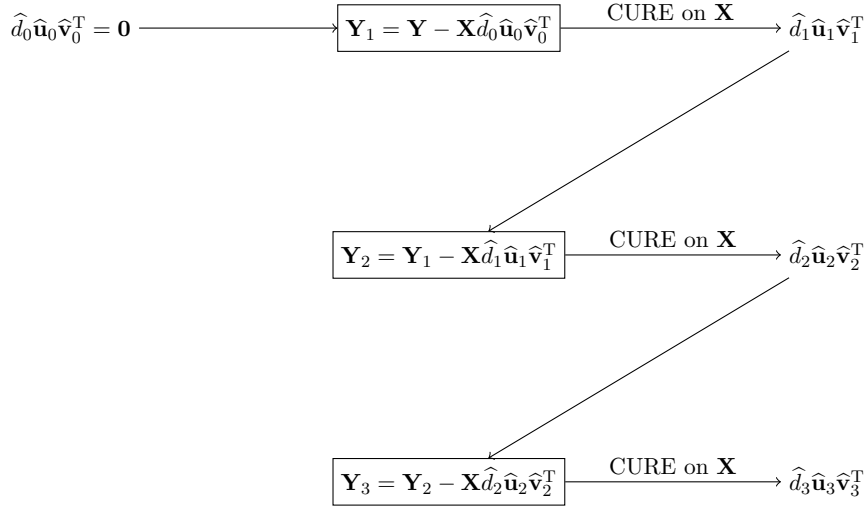

Figure 2: An illustration of the sequential pursuit.

Both deflation approaches are intuitive. The parallel pursuit uses an initial estimator to isolate the estimation of the unit-rank components, and refines the estimation of each component around

6

the vicinity of its initial estimator. Sequential pursuit, on the other hand, keeps extracting unit-rank components from the current residuals until no signal is left; the idea is related to a well-known fact derived from Eckart-Young Theorem (Eckart and Young, 1936) in matrix approximation and reduced-rank regression, that is, the solution of $\min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{XC}\|_F^2$ s.t. $\mathrm{rank}(\mathbf{C}) \leq r$ can be exactly recovered by sequentially fitting r unit-rank problems with the residuals from the previous step. But apparently, such exact correspondence would break down with additional sparse regularization.

Indeed, many matrix decomposition related methods have been proposed as repeatedly fitting rank-one models; see, e.g., Witten et al. (2009), Lee et al. (2010), and Ahn et al. (2015). Despite their proven effectiveness, many properties of these deflation strategies are yet to be explored. For example, what is the role of the initial estimator in parallel pursuit? Can parallel pursuit lead to substantially improved statistical performance (in terms of lowered error bound) comparing to that of the initial estimator? If Eckart-Young Theorem no longer applies, can sequential pursuit still work in the presence of regularization? What is the impact of "noise accumulation" since the subsequent estimation depends on the previous steps? How do the two deflation methods compare? We attempt to fill these gaps by performing a rigorous theoretical analysis, to be presented in the next section.

2.3 Unveiling the Mystery of Deflation

We now analyze the statistical properties of the deflation-based estimators of the sparse SVD components, i.e., $\widehat{\mathbf{C}}_k$, $k = 1, \ldots, r$, from either the parallel pursuit in (5) or the sequential pursuit in (6). Our treatment on the computation and the tuning of the CURE problem in (3) will be deferred to Section 3, as the second stage of our strategy. For now, we first concern theoretical analysis assuming CURE is solved globally, and then generalize our results to any computable local optimizer of CURE.

2.3.1 Technical Conditions

For any p by q nonzero matrix $\boldsymbol{\Delta}$, denote by $\boldsymbol{\Delta}_J$ the corresponding matrix of the same dimension that keeps the entries of $\boldsymbol{\Delta}$ with indices in set J while sets the others to be zero. We need the following regularity conditions.

Condition 2.1 *The random noise vectors are independently and identically distributed as $\mathbf{e}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, $i = 1, \ldots, n$. Denote the jth diagonal entry of $\boldsymbol{\Sigma}$ as σ_j^2; we assume $\sigma_{\max}^2 = \max_{1 \leq j \leq p} \sigma_j^2$ is bounded from above.*

Condition 2.2 *For $1 \leq k \leq r^*$, the gaps between the successive singular values are positive, i.e., $\delta_k^* = d_k^* - d_{k+1}^* > 0$, where $d_k^* > 0$ is the kth singular value of $n^{-1/2}\mathbf{XC}^*$.*

Condition 2.3 *There exists certain sparsity level s with a positive constant ρ_l such that*

$$\inf_{\boldsymbol{\Delta}} \left\{ \frac{\|\mathbf{X}\boldsymbol{\Delta}\|_F^2}{n(\|\boldsymbol{\Delta}_J\|_F^2 \vee \|\boldsymbol{\Delta}_{J_s^c}\|_F^2)} : |J| \leq s, \ \|\boldsymbol{\Delta}_{J^c}\|_1 \leq 3\|\boldsymbol{\Delta}_J\|_1 \right\} \geq \rho_l,$$

where $\boldsymbol{\Delta}_{J_s^c}$ is formed by keeping the s entries of $\boldsymbol{\Delta}_{J^c}$ with largest absolute values and setting the others to be zero.

7

Condition 2.4 *There exists certain sparsity level s with a positive constant ϕ_u such that*

$$\sup_{\mathbf{\Delta}} \left\{ \frac{|J|^{1/2}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{\Delta}\|_{\max}}{n\|\mathbf{\Delta}_J\|_F} : |J| \leq s, \ \|\mathbf{\Delta}_{J^c}\|_1 \leq 3\|\mathbf{\Delta}_J\|_1 \right\} \leq \phi_u.$$

Condition 2.1 assumes the random errors in model (1) are Gaussian for simplicity. In fact, our technical argument still applies as long as the tail probability bound of the noise decays exponentially; see the inequality (25) in the proof of Lemma 2. Condition 2.2 is imposed to ensure the identifiability of the r^* latent factors (singular vectors). Otherwise, the targeted unit rank matrices would not be distinguishable. Similar assumptions can be found in Mishra et al. (2017) , Zheng et al. (2021), and Zheng et al. (2019), among others.

Condition 2.3 is a matrix version of the restricted eigenvalue (RE) condition proposed in Bickel et al. (2009), which is typically imposed in ℓ_1-penalization to restrict the correlations between the columns of $\mathbf{X}$ within certain sparsity level, thus guaranteeing the identifiability of the true regression coefficients. The only difference is that we consider the estimation of coefficient matrices here instead of vectors. Since the Frobenius norm of a matrix can be regarded as the ℓ_2-norm of the stacked vector consisting of the columns of the matrix, Condition 2.3 is equivalent to the RE condition in the univariate response setting. The integer s here acts as a theoretical upper bound on the sparsity level of the true coefficient matrices, the requirement on which will be shown to be different in the two deflation approaches.

Similar to Condition 2.3, Condition 2.4 is a matrix version of the cone invertibility factor (Ye and Zhang, 2010) type of condition. It allows us to control the entrywise estimation error by assuming that $n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{\Delta}\|_{\max}$ can be bounded by some multiple of the averaged error $\|\mathbf{\Delta}_J\|_F/|J|^{1/2}$ when $\mathbf{\Delta}$ is restricted in the cone. In this way, the impact of the estimation error of the initial lasso estimate can spread out on the support instead of massing at one component.

2.3.2 Main Results

Now we are ready to present the main results. Denote by $s_k = \|\mathbf{C}_k^*\|_0$ for $1 \leq k \leq r^*$ and $s_0 = \|\mathbf{C}^*\|_0$. Denote $\widehat{\mathbf{\Delta}}_k = \widehat{\mathbf{C}}_k - \mathbf{C}_k^*$, for $k = 1, \ldots, r^*$. The following theorem characterizes the estimation accuracy in different layers of the sequential pursuit.

Theorem 1 (Convergence rates of the sequential pursuit) *Suppose Conditions 2.1–2.4 hold with the sparsity level $s \geq \max_{1 \leq k \leq r^*} s_k$. Choose $\lambda_1 = 2\sigma_{\max}\sqrt{2\alpha \log(pq)/n}$ for some constant $\alpha > 1$ and $\lambda_k = \Pi_{\ell=1}^{k-1}(1 + \eta_\ell)\lambda_1$. The following results hold uniformly over $1 \leq k \leq r^*$ with probability at least $1 - (pq)^{1-\alpha}$,*

$$\|\widehat{\mathbf{\Delta}}_k\|_F = O(\theta_k\sqrt{s_k}\lambda_k) = O(\gamma_k\theta_k\sqrt{s_k\log(pq)/n}),$$
$$\|\widehat{\mathbf{\Delta}}_k\|_1 = O(\theta_k s_k\lambda_k) = O(\gamma_k\theta_k s_k\sqrt{\log(pq)/n}),$$

where $\theta_k = d_k^*/\delta_k^*$, $\eta_k = c\theta_k$ with constant $c = 24\phi_u\rho_l^{-1}$, and $\gamma_k = \Pi_{\ell=1}^{k-1}(1 + \eta_\ell)$ with $\gamma_1 = 1$.

Theorem 1 presents the estimation error bounds in terms of both Frobenius and ℓ_1 norms for the co-sparse unit rank matrix estimators in sequential pursuit under mild and reasonable conditions. This is nontrivial since the subsequent layers play the role of extra noises in the estimation of each unit-rank matrix. We address this issue by showing that its impact is secondary due to the orthogonality between different layers, as demonstrated in Lemma 1. The results hold uniformly

over all true layers with a significant probability that approaches one in polynomial orders of the product of the two dimensions p and q.

The regularization parameter λ_k reflects the minimum penalization level needed to suppress the noise in the kth layer, which consists of two parts. One is from the random noise accompanied with the original response variables, while the other is due to the accumulation of the estimation errors from all previous layers. The latter is of a larger magnitude than the former one such that the penalization level increases almost exponentially with k. This can be inevitable in the sequential procedure since the kth layer estimator is based on the residual response matrix after extracting the previous layers. Hopefully, the estimation consistency is generally guaranteed for all the significant unit-rank matrices as long as the true regression coefficient matrix $\mathbf{C}^*$ is of sufficiently low rank.

When the singular values are well separated such that the magnitudes of the eigengaps δ_k^* are about the same as that of the eigenvalues, θ_k and η_k will be around a constant level. Then for the first few layers, the estimation accuracy is about $O(\sqrt{s_k \log(pq)/n})$, which is close to the optimal rate $O(\sqrt{(s_u + s_v) \log(p \vee q)/n})$ for the estimation of $\mathbf{C}^*$ established in Ma et al. (2020). When the true rank is one, the main difference between the two rates lies in the sparsity factors, where $(s_u + s_v)$ is a sum of the sparsity levels of the left and right singular vectors, while our sparsity factor is the product of them. However, the optimal rate is typically attained through some nonconvex algorithms (Ma et al., 2020; Yu et al., 2020; Uematsu et al., 2019) that search the optimal solution in a neighborhood of $\mathbf{C}^*$. In contrast, our CURE algorithm enjoys better computational efficiency and stability.

Last but not least, in view of the correlation constraint on the design matrix $\mathbf{X}$, that is, $s \geq \max_{1 \leq k \leq r^*} s_k$, the sequential pursuit is valid as long as the number of nonzero entries in each layer $\mathbf{C}_k^*$ is within the sparsity level s imposed in Conditions 2.3 and 2.4. It means that in practice, the sequential deflation strategy can recover some complex networks layer by layer, which is not shared by either the parallel pursuit or other methods that directly target on estimating the multi-rank coefficient matrix.

We then turn our attention to the statistical properties of the parallel pursuit. The key point of this deflation strategy is to find a relatively accurate initial estimator such that the signals from the other layers/components except the targeted one can be approximately removed. The reduced-rank regression estimator may not be a good choice under high dimensions, since it does not guarantee the estimation consistency due to the lack of sparsity constraint. Similar to Uematsu et al. (2019), we mainly adopt the following lasso initial estimator,

$$\widetilde{\mathbf{C}} = \arg \min_{\mathbf{C}} \ (2n)^{-1}\|\mathbf{Y} - \mathbf{X}\mathbf{C}\|_F^2 + \lambda_0\|\mathbf{C}\|_1,$$

which can be efficiently solved by various algorithms (Friedman et al., 2010). Under the same RE condition (Condition 2.3) as the sequential pursuit on the design matrix $\mathbf{X}$ with a tolerated sparsity level $s \geq s_0$, where s_0 indicates the number of nonzero entries in $\mathbf{C}^*$, we show that this lasso initial estimator is consistent with the convergence rate of $O(\sqrt{s_0 \log(pq)/n})$ in Lemma 3.

Based on $\widetilde{\mathbf{C}}$, we can obtain the initial unit rank estimates $\widetilde{\mathbf{C}}_k^0$ for different layers through a $\mathbf{P}$-orthogonal SVD. However, these unit rank estimates are not guaranteed to be sparse even if the lasso estimator $\widetilde{\mathbf{C}}$ is a sparse one, which causes additional difficulties in high dimensions. Thus, to facilitate the theoretical analysis, our initial kth layer estimate $\widetilde{\mathbf{C}}_k$ takes the s largest components of $\widetilde{\mathbf{C}}_k^0$ in terms of absolute values while sets the others to be zero, where s is the upper bound on the sparsity level defined in Conditions 2.3 and 2.4. We will show that these s-sparse initial

estimates $\widetilde{\mathbf{C}}_k$ can maintain the estimation accuracy of the initial unit-rank estimates $\widetilde{\mathbf{C}}_k^0$ without imposing any signal assumption, regardless of the thresholding procedure. In practice, we can use the SVD estimates directly as the impact of fairly small entries is negligible.

Theorem 2 (Convergence rates of the parallel pursuit) *Suppose Conditions 2.1–2.4 hold with the sparsity level $s \geq \max_{0 \leq k \leq r^*} s_k$. Choose $\lambda_0 = 2\sigma_{\max}\sqrt{2\alpha \log(pq)/n}$ with constant $\alpha > 1$ for the initial Lasso estimator $\widetilde{\mathbf{C}}$, and $\lambda_k = 2C\psi_k\sqrt{\log(pq)/n}$ for some positive constant C. The following results hold uniformly over $1 \leq k \leq r^*$ with probability at least $1 - (pq)^{1-\alpha}$,*

$$\|\widehat{\boldsymbol{\Delta}}_k\|_F = O(\sqrt{s_k}\lambda_k) = O(\psi_k\sqrt{s_k \log(pq)/n}),$$
$$\|\widehat{\boldsymbol{\Delta}}_k\|_1 = O(s_k\lambda_k) = O(\psi_k s_k\sqrt{\log(pq)/n}),$$

where $\psi_k = d_1^* d_c^*/(d_k^* \min[\delta_{k-1}^*, \delta_k^*])$ with d_c^* the largest singular value of $\mathbf{C}^*$.

Based on the lasso initial estimator, Theorem 2 demonstrates that the parallel pursuit achieves accurate estimation of different layers and reduces the sparsity factor in the convergence rates from s_0 to s_k, where the sparsity level s_0 of the entire regression coefficient matrix can be about r^* times of s_k. Moreover, as there is no accumulated noises in the parallel estimation of different layers, we do not see an accumulation factor γ_k here such that the penalization parameters keep around a uniform magnitude. The eigen-factor ψ_k plays a similar role as θ_k in the convergence rates for sequential pursuit in Theorem 1, both of which can typically be bounded from above under the low-rank and sparse structures. Thus, the estimation accuracy of all the significant unit-rank matrices is about the same as that of the first layer in the sequential pursuit, which reveals the potential superiority of the parallel pursuit.

In view of the technical assumptions, the sequential and the parallel pursuits are very similar, while the main difference lies in the requirement on the sparsity level s. As discussed after Condition 2.3, the integer s constrains the correlations between the columns of $\mathbf{X}$ to ensure the identifiability of the true supports, thus can be regarded as fixed for a given design matrix. Since the parallel pursuit requires not only the sparsity level s_k of each individual layer but also the overall sparsity level s_0 to be no larger than s, it puts a stricter constraint on correlations among the predictors. In other words, when the true coefficient matrix $\mathbf{C}^*$ is not sufficiently sparse, the lasso initial estimator may not be accurate enough to facilitate the subsequent parallel estimation, in which case the sequential pursuit could enjoy some advantage.

Our theoretical results mainly concern the convergence rate of estimating each unit-rank layer $\mathbf{C}_k^*$ by the estimator $\widehat{\mathbf{C}}_k$, for $1 \leq k \leq r^*$, where r^* is not required to be known when establishing the convergence rate of each $\widehat{\mathbf{C}}_k$. When $k > r^*$, there is no signal left in the residual matrix and the singular vectors $\mathbf{v}_k^*$ and $\mathbf{u}_k^*$ become unidentifiable due to the repeated zero singular values. In practice, the rank of the overall coefficient matrix can be naturally reduced or determined when some sparse unit-rank components are estimated to be zero by tuning λ via cross validation or some information criterion. In fact, r^* can also be detected before solving the CURE problems. Specifically, since r^* corresponds to the SVD of the regression component $\mathbf{X}\mathbf{C}^*$, tuning the optimal rank from the sequence of SVD components of $\mathbf{Y}$ by some information criterion has been shown to enjoy consistency guarantee under reasonable signal conditions (Zheng et al., 2022). Thus, we can identify the rank correspondingly before conducting the deflation strategies to help prevent overfitting from the residual matrix.

The previous two theorems guarantee the statistical properties of the global optimizers of the CURE problems from the two different deflation strategies, respectively. It is worth noticing that once the rates of convergence of $\mathbf{C}_k^*$ are established, the estimation errors of the left and the right singular vectors, as well as the singular values, can be shown to be around the same order since they are bounded from above by that of $\mathbf{C}_k^*$. Then the rates of convergence for $\mathbf{D}^*$, $\mathbf{U}^*$, and $\mathbf{V}^*$ will be no more than the aggregation of the estimation errors from the top r^* layers.

On the other hand, although the CURE problem adopts a bi-convex form, it is generally not convex such that the computational solution is usually a local optimizer rather than a global one. Fortunately, it has been proved in univariate response settings that any two sparse local optimizers can be close to each other under mild regularity conditions (Zhang and Zhang, 2012; Fan and Lv, 2014). Therefore, when the global minimizer is also sparse, the computational solution will share similar asymptotic properties. We illustrate this phenomenon in multi-response settings through the following theorem.

Theorem 3 *For any k, $1 \leq k \leq r^*$, let $\widehat{\mathbf{C}}_k^L$ be a computable local optimizer which satisfies $\|\widehat{\mathbf{C}}_k^L\|_0 = O(s_k)$ and $n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y}_k - \mathbf{X}\widehat{\mathbf{C}}_k^L)\|_{\max} = O(\lambda_k)$, $\min_{\|\boldsymbol{\gamma}\|_2=1,\|\boldsymbol{\gamma}\|_0 \leq Cs_k} n^{-1/2}\|\mathbf{X}\boldsymbol{\gamma}\|_2 \geq \kappa_0$ for some positive constant κ_0 and sufficiently large positive constant C, and $\|\widehat{\mathbf{C}}_k\|_0 = O(s_k)$. Then under the same assumptions of Theorem 1 (Theorem 2), $\widehat{\mathbf{C}}_k^L$ achieves the same estimation error bounds as the global optimizer.*

The regularity conditions on the computable local optimizer and the design matrix are basically the same as those in Fan and Lv (2014). The assumption on the sparsity of the global optimizer is generally easy to satisfy as we adopt the ℓ_1-penalization, which was shown in Bickel et al. (2009) to generate a sparse model of size $O(\phi^{\max}s)$ under certain regularity conditions, where $\phi^{\max}$ is the largest eigenvalue of the Gram matrix and s is the true sparsity level. For the local minimizer $\widehat{\mathbf{C}}_k^L$, the sparsity holds naturally because we update the solution restricted in an active set, where the number of nonzeros in the algorithm can be exactly controlled. Furthermore, the correlation between the predictors and the residuals is an assumption that we expect the solution from the algorithm can fit the responses well. In Lemma 5 and Theorem 6, we will show that the algorithmic solution can converges to ACS's with small enough stepsize. Thus, in view of Theorem 3, the sparse computational solutions we obtain in practice can satisfy the desirable statistical properties.

3. Contended Stagewise Learning

3.1 Stagewise CURE

It remains to solve the CURE problem in (3) or (4). The alternating convex search (ACS) algorithm, i.e., block coordinate descent (Minasian et al., 2014), is natural and commonly-used for solving (4), in which the objective function is alternately optimized with respect to a (overlapping) block of parameters, $(d, \mathbf{u})$ or $(d, \mathbf{v})$, with the rest held fixed. Since the objective is a function of $(d, \mathbf{u})$ or $(d, \mathbf{v})$ only through the products $d\mathbf{u}$ or $d\mathbf{v}$, the norm constraints are avoided in the sub-routines which then become exactly lasso problems. Mishra et al. (2017) showed that ACS converges to a coordinate-wise minimum point of (4) and the sequence of solutions along the iterations is uniformly bounded. With such a typical approach, however, the optimization needs to be repeated for a grid of λ values, and even with a warm-start strategy, the computation can still be expensive.

Inspired by the stagewise learning paradigm (Efron et al., 2004; Zhao and Yu, 2013; Tibshirani, 2015), we innovate a *contended stagewise learning strategy* to trace out the entire solution paths of CURE. Before we dive into details of our proposed algorithms, it is helpful to have an overview on the main ideas and the potential benefits of stagewise learning. Roughly speaking, stagewise learning is a slow-brewing process for model building. A stagewise procedure starts with a null model and takes simple, incremental steps to gradually update the parameters to enrich the model, in which each incremental update is determined by minimizing certain loss function over a simplified, restricted parameter space resulting from constraining either the form of the potential update or the size of the potential increase in some model complexity measure. As such, a carefully crafted stagewise procedure can efficiently trace out a whole spectrum of potential models with repetitive simple calculations. Stagewise learning has several prominent features that makes it very attractive in dealing with large-scale problems. First, it allows for a flexible trade-off between statistical accuracy and computational efficiency by the choice of the step size. Second, its learning process can stop early whenever a desirable model complexity is reached which avoids fitting overly complex models. Last but not the least, it admits a principled connection with regularized estimation, i.e., it has been shown for several problems that we can design a stagewise procedure to approximately trace out the solution paths of a corresponding regularized estimation problem, and the convergence of the paths could be exact when the step size goes to zero. However, thur far, stagewise learning is still largely heuristic and mainly limited to convex problems such as lasso, group lasso or nuclear norm penalized regression (Tibshirani, 2015; Vaughan et al., 2017).

Although CURE is non-convex, we come to realize that efficient and principled stagewise learning remains possible. Our idea is simple yet elegant: when determining the update at each stagewise step, all the proposals from the subproblems of the potential blockwise updates have to compete with each other, and only the winner gets executed. This is in contrast to the classical setup, where in each update there is only a single proposal that is determined in a global fashion.

We now present in detail our proposed stagewise procedure for CURE. For improving the estimation stability and facilitate calculations in practice, we consider a more general CURE problem with an added ℓ_2 penalty term and adopt an alternative set of normalization constraints,

$$\min_{d,\mathbf{u},\mathbf{v}} Q(d\mathbf{u}\mathbf{v}^{\mathrm{T}}; \lambda) = \left\{ L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) + \lambda\rho(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) \right\}, \qquad \text{s.t.}\, d \geq 0, \|\mathbf{u}\|_1 = 1, \|\mathbf{v}\|_1 = 1, \tag{7}$$

where $L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) = (2n)^{-1}\|\mathbf{Y} - d\mathbf{X}\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_F^2 + (\mu/2)\|d\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_F^2$ and $\rho(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) = \lambda d\|\mathbf{u}\|_1\|\mathbf{v}\|_1$. This is the same as using a strictly convex elastic net penalty on $\mathbf{C} = d\mathbf{u}\mathbf{v}^{\mathrm{T}}$ whenever $\mu > 0$ (Zou and Hastie, 2005). Here we consider μ as a pre-specified fixed constant, and our goal remains to trace out the solution paths with respect to varying λ. In Theorem 6, we show that the ℓ_2 term can asymptotically vanish, so the CURE problems with or without the ℓ_2 penalty are asymptotically equivalent. Since the objective is a function of $d\mathbf{u}\mathbf{v}^{\mathrm{T}}$, the normalization of $\mathbf{u}$ and $\mathbf{v}$ can be arbitrary and can always be absorbed to d; we thus use ℓ_1 normalization here, and the solution can be easily re-normalized to satisfy the original set of constraints in (4), corresponding to the $\mathbf{P}$-orthogonal SVD.

In what follows, we first present the general structure of the contended stagewise learning algorithm, and then provide the resulting simple problems/solutions for the incremental updates. The detailed derivations are given in Appendix D.

(I) Initialization. Set $t = 0$. This step is based on searching for the initial non-zero entry of $\mathbf{C}$ as follows:

$$
\begin{aligned}
(\widehat{j}, \widehat{k}, \widehat{s}) &= \arg \min_{(j,k); s = \pm\epsilon} L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) \text{ s.t. } du_j v_k = s, u_{j'} v_{k'} = 0, \forall j' \neq j, k' \neq k; \\
&= \arg \min_{(j,k); s = \pm\epsilon} L(s\mathbf{1}_j \mathbf{1}_k^{\mathrm{T}}); \\
\widehat{\mathbf{u}}^0 &= \mathbf{1}_{\widehat{j}}; \widehat{\mathbf{v}}^0 = \operatorname{sgn}(\widehat{s})\mathbf{1}_{\widehat{k}}; \widehat{d}^0 = \epsilon; \\
\mathcal{A}^0 &= \{\widehat{j}\}, \mathcal{B}^0 = \{\widehat{k}\}; \lambda^0 = \frac{1}{\epsilon}\{L(\mathbf{0}) - L(\widehat{d}^0 \widehat{\mathbf{u}}^0 \widehat{\mathbf{v}}^{0\mathrm{T}})\}.
\end{aligned}
\tag{8}
$$

Here $\mathbf{1}_j$ ($\mathbf{1}_k$) is an $p \times 1$ ($q \times 1$) standard basis vector with all zeros except for a one in its jth (kth) coordinate. The $\mathcal{A}^0$ and $\mathcal{B}^0$ are the initial row and column active sets of non-zero indices, respectively.

(II) Backward Update. The update is about searching for the best row or column index to move "backward" within the current active sets in a blockwise fashion. At the $(t+1)$th step, there are two options:

$$
\begin{aligned}
\widehat{j} &= \arg \min_{j \in \mathcal{A}^t} L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) \text{ s.t. } (d\mathbf{u}) = (d\mathbf{u})^t - \operatorname{sgn}(u_j^t)\epsilon\mathbf{1}_j, \mathbf{v} = \mathbf{v}^t; \\
(d\mathbf{u})^{t+1} &= (d\mathbf{u})^t - \operatorname{sgn}(u_{\widehat{j}}^t)\epsilon\mathbf{1}_{\widehat{j}}; d^{t+1} = \|(d\mathbf{u})^{t+1}\|_1; \mathbf{u}^{t+1} = (d\mathbf{u})^{t+1}/d^{t+1}; \mathbf{v}^{t+1} = \mathbf{v}^t; \text{(scaling)} \\
\mathcal{A}^{t+1} &= \mathcal{A}^t \text{ if } u_{\widehat{j}}^{t+1} \neq 0; \mathcal{A}^{t+1} = \mathcal{A}^t - \{\widehat{j}\} \text{ if } u_{\widehat{j}}^{t+1} = 0; \mathcal{B}^{t+1} = \mathcal{B}^t; \\
\lambda^{t+1} &= \lambda^t;
\end{aligned}
\tag{9}
$$

or

$$
\begin{aligned}
\widehat{k} &= \arg \min_{k \in \mathcal{B}^t} L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) \text{ s.t. } (d\mathbf{v}) = (d\mathbf{v})^t - \operatorname{sgn}(v_k^t)\epsilon\mathbf{1}_k, \mathbf{u} = \mathbf{u}^t; \\
(d\mathbf{v})^{t+1} &= (d\mathbf{v})^t - \operatorname{sgn}(v_{\widehat{k}}^t)\epsilon\mathbf{1}_{\widehat{k}}; d^{t+1} = \|(d\mathbf{v})^{t+1}\|_1; \mathbf{v}^{t+1} = (d\mathbf{v})^{t+1}/d^{t+1}; \mathbf{u}^{t+1} = \mathbf{u}^t; \\
\mathcal{A}^{t+1} &= \mathcal{A}^t; \mathcal{B}^{t+1} = \mathcal{B}^t \text{ if } v_{\widehat{k}}^{t+1} \neq 0; \mathcal{B}^{t+1} = \mathcal{B}^t - \{\widehat{k}\} \text{ if } v_{\widehat{k}}^{t+1} = 0; \\
\lambda^{t+1} &= \lambda^t.
\end{aligned}
\tag{10}
$$

The penalty term will be decreased by a fixed amount $\lambda^t\epsilon$ (as $\|\mathbf{u}^t\|_1 = \|\mathbf{v}^t\|_1 = 1$); the option with a lower L value wins the bid. To ensure the objective in (4) is reduced, the winning option is executed only if $L(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) < \lambda^t\epsilon - \xi$, where $\xi = o(\epsilon) > 0$ is a tolerance level.

(III) Forward Update. When the backward update can no longer proceed, a forward update is carried out, which searches for the best row or column index to move "forward" over all indices, again, in a blockwise fashion. At the $(t+1)$th step, the update chooses between two proposals:

$$
\begin{aligned}
(\widehat{j}, \widehat{s}) &= \arg \min_{j; s = \pm\epsilon} L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) \text{ s.t. } (d\mathbf{u}) = (d\mathbf{u})^t + s\mathbf{1}_j, \mathbf{v} = \mathbf{v}^t; \\
(d\mathbf{u})^{t+1} &= (d\mathbf{u})^t + \widehat{s}\mathbf{1}_{\widehat{j}}; d^{t+1} = \|(d\mathbf{u})^{t+1}\|_1; \mathbf{u}^{t+1} = (d\mathbf{u})^{t+1}/d^{t+1}; \mathbf{v}^{t+1} = \mathbf{v}^t; \\
\mathcal{A}^{t+1} &= \mathcal{A}^t \cup \widehat{j}, \mathcal{B}^{t+1} = \mathcal{B}^t; \\
\lambda^{t+1} &= \min(\lambda^t, \{L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) - L(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \xi\}/\epsilon),
\end{aligned}
\tag{11}
$$

or

$$(\widehat{k}, \widehat{h}) = \arg \min_{k; h = \pm \epsilon} L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) \text{ s.t. } (d\mathbf{v}) = (d\mathbf{v})^t + h\mathbf{1}_k, \mathbf{u} = \mathbf{u}^t;$$

$$(d\mathbf{v})^{t+1} = (d\mathbf{v})^t + \widehat{h}\mathbf{1}_{\widehat{k}}; d^{t+1} = \|(d\mathbf{v})^{t+1}\|_1; \mathbf{v}^{t+1} = (d\mathbf{v})^{t+1}/d^{t+1}; \mathbf{u}^{t+1} = \mathbf{u}^t;$$

$$\mathcal{A}^{t+1} = \mathcal{A}^t, \mathcal{B}^{t+1} = \mathcal{B}^t \cup \widehat{k};$$

$$\lambda^{t+1} = \min(\lambda^t, \{L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) - L(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \xi\}/\epsilon).$$

(12)

As $\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \rho(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) = \epsilon$, the option that can decrease L more is executed.

The structure of the proposed stagewise learning procedure is summarized in Algorithm 1. It is intuitive how this procedure works. A forward update is taken only when backward updates can no longer proceed, i.e., the regularized loss at λ^t can no longer be reduced by searching over the "backward directions" in a blockwise fashion within the current active sets. When a forward step is taken, the search expends to all possible blockwise directions, and the parameter λ^t gets reduced only when the corresponding regularized loss can not be further reduced by any incremental update on any block of parameters. Indeed, we show in Section 3.2 that when ϵ and $\xi = o(\epsilon)$ go to zero at the time λ^t gets reduced to λ^{t+1}, a coordinatewise minimum point of the regularized loss with λ^t can be reached.

Algorithm 1 Contended Stagewise Learning for CURE (Pseudo code)

Initialization step: based on searching for the first nonzero entries in $\mathbf{u}$ and $\mathbf{v}$ by (8). Set ϵ and ξ. Set $t = 0$.

repeat

    Backward step: compare the proposals of blockwise updates within the current active sets, and choose the one that induces the least increment in L (or the most reduction):

        • Proposal 1: update $(d, \mathbf{u})$, $\mathcal{A}^{t+1}$ and λ^{t+1} by (9).

        • Proposal 2: update $(d, \mathbf{v})$, $\mathcal{B}^{t+1}$ and λ^{t+1} by (10).

    if the induced increment in L is less than $\lambda^t\epsilon - \xi$, **then**

        Execute the chosen proposal of backward update.

    else

        Forward step: compare the proposals of blockwise updates over all row/column indices, choose the one that reduces the loss L the most:

        • Proposal 1: update $(d, \mathbf{u})$, $\mathcal{A}^{t+1}$ and λ^{t+1} by (11).

        • Proposal 2: update $(d, \mathbf{v})$, $\mathcal{B}^{t+1}$ and λ^{t+1} by (12).

    $t \to t + 1$.

until $\lambda_t \leq 0$.

The computations involved in the contended stagewise learning can be implemented in a very efficient way. Define the current residual matrix as $\mathbf{E}^t = \mathbf{Y} - d^t\mathbf{X}\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}$ and denote its kth column as $\widetilde{\mathbf{e}}_k$. Recall that $\widetilde{\mathbf{x}}_j$ and $\widetilde{\mathbf{y}}_k$ are the jth and kth column of $\mathbf{X}$ and $\mathbf{Y}$, respectively, as defined in Section 2. Then, the initialization step boils down to the following,

$$(\widehat{j}, \widehat{k}) = \arg \min_{j,k} \left\{ (2n)^{-1}\epsilon\|\widetilde{\mathbf{x}}_j\|_2^2 - n^{-1}|\widetilde{\mathbf{x}}_j^{\mathrm{T}}\widetilde{\mathbf{y}}_k| \right\}, \qquad \widehat{s} = \mathrm{sgn}(\widetilde{\mathbf{x}}_j^{\mathrm{T}}\widetilde{\mathbf{y}}_k)\epsilon.$$

(13)

The problem of the $(t+1)$th backward step is

$$
\begin{aligned}
\widehat{j} &= \arg\min_{j \in \mathcal{A}^t} (2n)^{-1}\epsilon\|\widetilde{\mathbf{x}}_j\|_2^2\|\mathbf{v}^t\|_2^2 + n^{-1}\mathrm{sgn}(u_j^t)\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t - \mu d^t|u_j^t|\|\mathbf{v}^t\|_2^2, \\
\widehat{k} &= \arg\min_{k \in \mathcal{B}^t} n^{-1}\mathrm{sgn}(v_k^t)\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t - \mu d^t|v_k^t|\|\mathbf{u}^t\|_2^2.
\end{aligned}
\tag{14}
$$

The problem of the $(t+1)$th forward step is

$$
\begin{aligned}
\widehat{j} &= \arg\max_{j} |n^{-1}\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t - \mu d^t u_j^t\|\mathbf{v}^t\|_2^2| - (2n)^{-1}\epsilon\|\widetilde{\mathbf{x}}_j\|_2^2\|\mathbf{v}^t\|_2^2; \\
\widehat{s} &= \mathrm{sgn}(n^{-1}\widetilde{\mathbf{x}}_{\widehat{j}}^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t - \mu d^t u_{\widehat{j}}^t\|\mathbf{v}^t\|_2^2)\epsilon, \\
\widehat{k} &= \arg\max_{k} |n^{-1}\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t - \mu d^t v_k^t\|\mathbf{u}^t\|_2^2|; \\
\widehat{h} &= \mathrm{sgn}(n^{-1}\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_{\widehat{k}}^t - \mu d^t v_{\widehat{k}}^t\|\mathbf{u}^t\|_2^2)\epsilon.
\end{aligned}
\tag{15}
$$

Lastly, for choosing between updating $d\mathbf{u}$ or $d\mathbf{v}$ or between proceeding with backward or forward update, the change in the loss function are computed as

$$
\begin{aligned}
L\left(\left[(d\mathbf{u})^t + \widehat{s}\mathbf{1}_{\widehat{j}}\right]\mathbf{v}^{t\mathrm{T}}\right) - L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) &= \frac{\epsilon^2}{2n}\|\widetilde{\mathbf{x}}_{\widehat{j}}\|_2^2\|\mathbf{v}^t\|_2^2 - \frac{\widehat{s}}{n}\widetilde{\mathbf{x}}_{\widehat{j}}^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t + \frac{\mu}{2}\epsilon^2\|\mathbf{v}^t\|_2^2 + \mu\widehat{s}d^t u_{\widehat{j}}^t\|\mathbf{v}^t\|_2^2, \\
L\left(\mathbf{u}^t\left[(d\mathbf{v})^t + \widehat{h}\mathbf{1}_{\widehat{k}}\right]^{\mathrm{T}}\right) - L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) &= \frac{\epsilon^2}{2n}\|\mathbf{X}\mathbf{u}^t\|_2^2 - \frac{\widehat{h}}{n}\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_{\widehat{k}}^t + \frac{\mu\epsilon^2}{2}\|\mathbf{u}^t\|_2^2 + \mu\widehat{h}d^t v_{\widehat{k}}^t\|\mathbf{u}^t\|_2^2.
\end{aligned}
$$

To streamline the main ideas, we haven't discussed much on the handling of missing values. In fact, our stagewise methods can conveniently handle missing values in the response matrix, which makes it applicable for large-scale matrix completion (Candès and Recht, 2009; Candès and Plan, 2010). Here we briefly describe the setup. Let $\mathcal{H}$ be the index set of all observed values in $\mathbf{Y}$, i.e., $\mathcal{H} = \{(i, j); y_{ij}$ is observed, $i = 1, \ldots, n, j = 1, \ldots, q\}$. Define $P_{\mathcal{H}} : \mathbb{R}^{n \times q} \to \mathbb{R}^{n \times q}$ be the projection operator onto $\mathcal{H}$, so that for any matrix $\mathbf{Z} \in \mathbb{R}^{n \times q}$, the entries of $P_{\mathcal{H}}(\mathbf{Z})$ equal to those of $\mathbf{Z}$ on $\mathcal{H}$ and otherwise equal to zero. Then, corresponding to the complete data case in (7), the CURE problem with incomplete data can be expressed as

$$
\begin{aligned}
\min_{d,\mathbf{u},\mathbf{v}} \quad & (2n)^{-1}\|P_{\mathcal{H}}(\mathbf{Y}) - P_{\mathcal{H}}(d\mathbf{X}\mathbf{u}\mathbf{v}^{\mathrm{T}})\|_F^2 + \frac{\mu}{2}\|d\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_F^2 + \lambda d\|\mathbf{u}\|_1\|\mathbf{v}\|_1 \\
\text{s.t.} \quad & d \geq 0, \|\mathbf{u}\|_1 = 1, \ \|\mathbf{v}\|_1 = 1.
\end{aligned}
$$

We can show that the proposed methods still work with slight modification and maintain computational efficiency; the only change is to replace the residual matrix $\mathbf{E}^t$ with its projection $P_{\mathcal{H}}(\mathbf{E}^t)$ during the iterations.

We have implemented all the proposed computational methods in a user-friendly R package with `RCpp` (R Core Team, 2021).

3.2 Computational Complexity and Convergence

Theorem 4 presents the computational complexity of the proposed stagewise procedure.

Theorem 4 *Consider the $(t+1)$th step of Algorithm 1 for contended stagewise learning of* (7). *Let $\mathcal{A}^t = a^t$ and $\mathcal{B}^t = b^t$. The computational complexity of the update is $O(a^t nq + b^t np)$.*

The results show that the stagewise learning can be much more efficient than the ACS approach. ACS needs to be run for a grid of λ values; for each fixed λ, the algorithm alternates between two lasso problems with computational complexity $O(npq)$ (Friedman et al., 2010) until convergence. In contrast, our method costs $O(npq)$ operations in the initialization step and $O(a^t nq + b^t np)$ operations in the subsequent steps, and the solution paths are traced out in a single run.

We now attempt to quantify the proximity between the stagewise approximated solution and that from fully optimizing the regularized loss function through ACS.

Lemma 5 *Consider Algorithm 1 for the contended stagewise learning of (7). When $\lambda^{t+1} < \lambda^t$, that is, whenever λ^t gets reduced during the stagewise learning, we have*

$$\max \left[\|(d\mathbf{u})^{t*}\mathbf{v}^{t\mathrm{T}} - d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}\|_F, \ \|\mathbf{u}^t(d\mathbf{v})^{t*\mathrm{T}} - d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}\|_F \right] \leq 2\sqrt{pq} \left(\frac{M\epsilon}{2\mu} + \frac{\xi}{\epsilon\mu} \right),$$

where M is a constant, and

$$(d\mathbf{u})^{t*} = \arg\min_{d\mathbf{u}} \ Q(d\mathbf{u}\mathbf{v}^{t\mathrm{T}}; \lambda^t), \ (d\mathbf{v})^{t*} = \arg\min_{d\mathbf{v}} \ Q(\mathbf{u}^t(d\mathbf{v})^{\mathrm{T}}; \lambda^t).$$

Lemma 5 attempts to quantify and bound the distance between any stagewise solution along the path (for any λ^t when it gets reduced) and the corresponding coordinate-wise minimum point for the optimization problem in (7) with $\lambda = \lambda^t$; the latter can be achieved by ACS. Due to the bi-convex nature of the problem, the bound is given in a "bi-directional" or coordinate-wise fashion with respect to either $d\mathbf{u}^t$ or $d\mathbf{v}^t$. The most important finding is on the order of the difference, which provides guidelines on the choices of ϵ, ξ and μ. Specifically, let $\mathbb{B}(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; 2\sqrt{pq}[\frac{M\epsilon}{2\mu} + \frac{\xi}{\epsilon\mu}])$ be a $(pq-1)$-dimensional ball with center $d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}$ and radius $2\sqrt{pq}[\frac{M\epsilon}{2\mu} + \frac{\xi}{\epsilon\mu}]$. By Lemma 5, the center will converge to the corresponding solution of ACS when the radius converges to zero, because $(d\mathbf{u})^{t*}\mathbf{v}^{t\mathrm{T}}$ and $\mathbf{u}^t(d\mathbf{v})^{t*\mathrm{T}}$ are always in the ball. In addition, the inequality in Lemma 5 holds for any $t \geq 0$, so we can directly obtain the pathwise convergence as shown in the following theorem.

Theorem 6 *Consider Algorithm 1 for the contended stagewise learning of (7). When $\epsilon \to 0$ and $\xi = o(\epsilon)$, the solution paths of the stagewise learning converge to those of the ACS uniformly. Moreover, the results remain hold when $\mu \to 0$, $\epsilon = o(\mu)$ and $\xi = o(\epsilon\mu)$.*

3.3 Tuning & Early Stopping

In practice, it is often required to identify the optimal solution along the paths. The K-fold cross validation can be used to evaluate the models, for which the stagewise learning procedure has to be run multiple times. Alternatively, selecting tuning parameters by information criterion can be more efficient. In our implementation, the default choice is to use the generalized information criterion (GIC) (Fan and Tang, 2013):

$$\mathrm{GIC}(\lambda^t) = \log\|\mathbf{Y} - d^t\mathbf{X}\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}\|_F^2 + \frac{\mathrm{loglog}(nq)\log(pq)}{nq}\widehat{df}(\lambda^t)$$

where $\widehat{df}(\lambda^t) = \|\mathbf{u}^t\|_0 + \|\mathbf{v}^t\|_0 - 1$, is the estimated degrees of freedom for the stagewise model. GIC is shown to perform well in our numerical studies. Other implemented criteria including AIC, BIC, and out-of-sample prediction error (when additional testing samples are available) are optional.

Due to the nature of stagewise learning, an early stopping mechanism can be implemented based on monitoring the information criterion during the learning process. As the model complexity gradually increases during the stagewise learning, the information criterion, which balances model fitting and model complexity, is expected to first decrease and then increase. As such, the stagewise learning can be terminated early if such a convex pattern of the information criterion is detected. In our implementation, the default is to stop the stagewise learning if the information criterion has not being decreasing for 300 consecutive steps. This strategy avoids the fitting of overly-complex models, and neither specifying nor tuning the iteration number is required.

4. Simulation

4.1 Setups

We conduct simulation studies with data generated from the co-sparse factor regression model as specified in (1). We consider three simulation setups, which differ mainly on the generation of the true coefficient matrix $\mathbf{C}^* = \mathbf{U}^* \mathbf{D}^* \mathbf{V}^{*\mathrm{T}} \in \mathbb{R}^{p \times q}$, where $\mathbf{U}^* = [\mathbf{u}_1^*, \ldots, \mathbf{u}_{r^*}^*]$, $\mathbf{V}^* = [\mathbf{v}_1^*, \ldots, \mathbf{v}_{r^*}^*]$, and $\mathbf{D}^* = \mathrm{diag}\{d_1^* \ldots, d_{r^*}^*\}$.

Model I is a unit-rank model mainly for analyzing the properties of CURE, in which we set

$$\mathbf{u}_1^* = \bar{\mathbf{u}}_1 / \|\bar{\mathbf{u}}_1\|_2 \text{ where } \bar{\mathbf{u}}_1 = [10, -10, 8, -8, 5, -5, \mathrm{rep}(3, 5), \mathrm{rep}(-3, 5), \mathrm{rep}(0, p - 16)]^{\mathrm{T}},$$

$$\mathbf{v}_1^* = \bar{\mathbf{v}}_1 / \|\bar{\mathbf{v}}_1\|_2 \text{ where } \bar{\mathbf{v}}_1 = [10, -9, 8, -7, 6, -5, 4, -3, \mathrm{rep}(2, 17), \mathrm{rep}(0, q - 25)]^{\mathrm{T}},$$

and $d_1^* = 20$, where $\mathrm{rep}(a, b)$ represent a $1 \times b$ vector with all entries equaling to a.

Models II and III are multi-rank models. In Model II, the singular values and singular vectors are generated as follows,

$$\mathbf{u}_k^* = \bar{\mathbf{u}}_k / \|\bar{\mathbf{u}}_k\|_2 \text{ where } \bar{\mathbf{u}}_k = [\mathrm{rep}(0, k - 1), \mathrm{unif}(\mathcal{Q}_u, s_u), \mathrm{rep}(0, p - s_u - k + 1)]^{\mathrm{T}},$$

$$\mathbf{v}_k^* = \bar{\mathbf{v}}_k / \|\bar{\mathbf{v}}_k\|_2 \text{ where } \bar{\mathbf{v}}_k = [\mathrm{rep}(0, k - 1), \mathrm{unif}(\mathcal{Q}_v, s_v), \mathrm{rep}(0, q - s_v - k + 1)]^{\mathrm{T}},$$

$$d_k^* = 5 + 5(r^* - k + 1),$$

for $k = 1, \ldots, r^*$, where $s_u = 3$, $s_v = 4$, and $\mathrm{unif}(\mathcal{Q}, s)$ denotes a vector of length s whose entries are i.i.d. uniformly distributed on the set $\mathcal{Q}$; here we set $\mathcal{Q}_u = \{1, -1\}$ and $\mathcal{Q}_v = [-1, -0.3] \cup [0.3, 1]$ and adopt an additional Gram-Schmidt orthogonalization on $\bar{\mathbf{v}}_k$ to ensure the orthogonality. Model III is similar to Model II, and the only difference is that we generate $\bar{\mathbf{u}}_k$ and $\bar{\mathbf{v}}_k$ by

$$\bar{\mathbf{u}}_k = [\mathrm{rep}(0, s_u(k - 1)), \mathrm{unif}(\mathcal{Q}_u, s_u), \mathrm{rep}(0, p - ks_u)]^{\mathrm{T}},$$

$$\bar{\mathbf{v}}_k = [\mathrm{rep}(0, s_v(k - 1)), \mathrm{unif}(\mathcal{Q}_v, s_v), \mathrm{rep}(0, q - ks_v)]^{\mathrm{T}}.$$

In view of $\mathbf{C}^* = \sum_{k=1}^{r^*} d_k^* \mathbf{u}_k^* \mathbf{v}_k^{*\mathrm{T}}$, the coefficient matrix in Model II is more sparse than that in Model III due to the overlap of nonzero components in different unit-rank matrices.

We then generate the design matrix $\mathbf{X}$, following the same procedure as in Mishra et al. (2017). Specifically, let $\mathbf{x} \sim N(\mathbf{0}, \mathbf{\Gamma})$, where $\mathbf{\Gamma} = (\gamma_{ij})_{p \times p}$ with $\gamma_{ij} = 0.5^{|i-j|}$. Given $\mathbf{U}^* = [\mathbf{u}_1^*, \ldots, \mathbf{u}_{r^*}^*]$, we can find $\mathbf{U}_\perp^* \in \mathbb{R}^{p \times (p - r^*)}$ such that $\mathbf{P} = [\mathbf{U}^*, \mathbf{U}_\perp^*] \in \mathbb{R}^{p \times p}$ and $\mathrm{rank}(\mathbf{P}) = p$. Denote $\mathbf{x}_1 = \mathbf{U}^{*\mathrm{T}} \mathbf{x}$ and $\mathbf{x}_2 = \mathbf{U}_\perp^{*\mathrm{T}} \mathbf{x}$. We first generate a matrix $\mathbf{X}_1 \in \mathbf{R}^{n \times r^*}$ whose entries are from $N(\mathbf{0}, \mathbf{I}_{r^*})$ and then we generate $\mathbf{X}_2 \in \mathbb{R}^{n \times (p - r^*)}$ by drawing n random samples from the conditional distribution of $\mathbf{x}_2$ given $\mathbf{x}_1$. The predictor matrix is then set as $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2] \mathbf{P}^{-1}$ so that the setup corresponds to (2). The rows of the error matrix $\mathbf{E}$ are generated as i.i.d. samples from $N(0, \sigma^2 \mathbf{\Delta})$ where $\mathbf{\Delta} = (\delta_{ij})_{q \times q}$ with $\delta_{ij} = \rho^{|i-j|}$. The response $\mathbf{Y}$ is then generate by $\mathbf{Y} = \mathbf{X}\mathbf{C}^* + \mathbf{E}$. We set σ to control the signal-to-noise ratio (SNR), defined as $\mathrm{SNR} = \|d_{r^*}^* \mathbf{X} \mathbf{u}_{r^*}^* \mathbf{v}_{r^*}^{*\mathrm{T}}\|_2 / \|\mathbf{E}\|_F$.
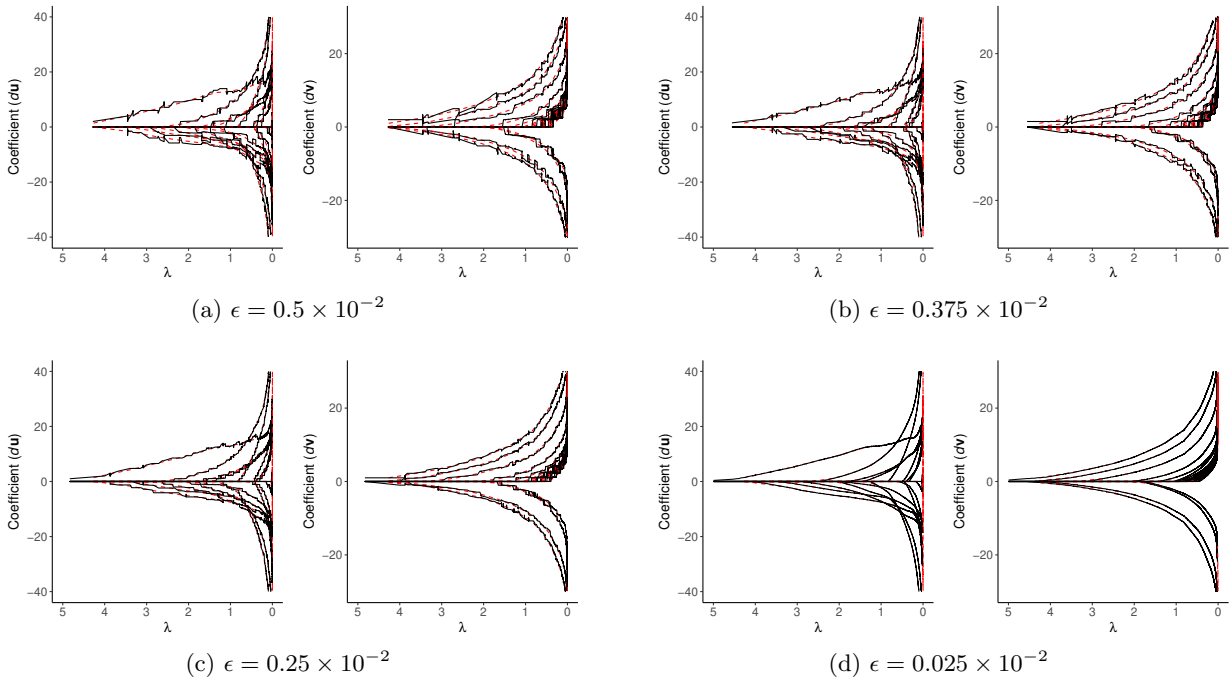
Figure 3: Stagewsie paths of $\widehat{d\mathbf{u}}$ and $\widehat{d\mathbf{v}}$ for different step sizes. Stagewise paths are shown as black solid lines, and the exact solution paths from ACS are shown in red dashed lines.

4.2 Path Convergence of Stagewise CURE

We demonstrate that the stagewise paths of CURE closely mimic the solutions of ACS as the step size becomes sufficiently small. We use Model I, with $n = p = q = 200$, $\rho = 0.3$ and SNR $= 0.25$. Figure 3 shows the solution paths of $\widehat{d\mathbf{u}}$ and $\widehat{d\mathbf{v}}$ with varying ϵ values. Indeed, the stagewise paths approximately trace out the solution paths of ACS, and their discrepancies vanish as the step size gets smaller.

4.3 Estimation Performance

We compare the estimation accuracy of our proposed approaches to several competing regularized regression methods, including reduced-rank regression (RRR), row-sparse reduced rank regression via adaptive group lasso (SRRR) (Chen and Huang, 2012), sparse orthogonal factor regression (SO-FAR) (Uematsu et al., 2019). For the two deflation based methods, we try both stagewise learning (STL) and ACS for solving CURE, resulting in sequential pursuit with stagewise learning (Se-qSTL), sequential pursuit with ACS (SeqACS), parallel pursuit with stagewise learning initialized by lasso (ParSTL(L)), parallel pursuit with stagewise learning initialized by RRR (ParSTL(R)), parallel pursuit with ACS initialized by lasso (ParACS(L)), and parallel pursuit with ACS initialized by RRR (ParACS(R)). The ParACS(R) and SeqACS are essentially corresponding to the reduced-rank regression with a sparse singular value decomposition (RSSVD) proposed by Chen et al. (2012) and the sequential co-sparse factor regression (SeFAR) proposed by Mishra et al. (2017), respectively.

Models II and III are considered here with $n = q = 100$, $p \in \{100, 200, 400\}$, $r^* \in \{3, 6\}$, SNR $\in \{0.25, 0.5, 1\}$, $\rho = 0.3$ and $\epsilon = 0.5 \times 10^{-2}$. The experiment under each setting is repeated 200 times. The estimation accuracy is measured by $\mathrm{Er}(\widehat{\mathbf{C}}) = \|\widehat{\mathbf{C}} - \mathbf{C}^*\|_F^2 / (pq)$ and $\mathrm{Er}(\mathbf{X}\widehat{\mathbf{C}}) = \|\mathbf{X}(\widehat{\mathbf{C}} - \mathbf{C}^*)\|_F^2 / (nq)$. The variable selection performance is characterized by the false positive rate (FPR) and false negative rate (FNR) in recovering the sparsity patterns of the SVD structure, where FPR = FP/(TN + FP) and FNR = FN/(TP + FN). Here, TP, FP, TN, and FN are the numbers of true nonzeros, false nonzeros, true zeros, and false zeros of $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$, respectively. Table 1 reports the results for Model II with SNR = 0.5 and $p \in \{200, 400\}$, and Table 2 reports the results for Model III under the same settings.

First of all, the stagewise methods are much faster than the other methods, and the efficiency gain in computation can be dramatic for models of large dimensions; more results regarding computation time can be found in Section 4.5. In general the estimation and prediction performance of stagewise method are better or comparable to the other competing methods. RRR performs the worst, as it does not consider sparse estimation at all. SOFAR may perform unsatisfactorily because its optimization is highly non-convex and it targets on the sparsity pattern in the SVD of $\mathbf{C}^*$ rather than its $\mathbf{P}-$orthogonal SVD. SRRR considers row-wise sparsity in $\mathbf{C}^*$ only. In terms of the ACS solutions for CURE, with a more sparse $\mathbf{C}^*$, parallel pursuit usually performs better than sequential pursuit, which is consistent with our theoretical results. When $\mathbf{C}^*$ becomes less sparse, the difference between parallel and sequential methods becomes negligible. It is also interesting to notice that stagewise methods may even slightly outperform their ACS counterparts. This may be due to the additional regularization effects of the stagewise approximation.

4.4 Impact of Step Size

We investigate the impact of the step size ϵ in stagewise learning on the performance of the proposed methods. The data are simulated from Model III with $p = 600$, $q = n = 200$, $r^* = 3$ and $\rho = 0.3$. We consider different step sizes, i.e., ϵ in $\{0.125, 0.25, 0.375, 0.5, 0.625\} \times 10^{-2}$. The experiment is replicated 200 times. Figure 4 shows the boxplots of the estimation error, prediction error, and computation time. The performance of the stagewise methods is stabilized when ϵ is small enough, i.e., $\epsilon \leq 0.25 \times 10^{-2}$ in this example. But the computational cost will increase if ϵ is too small and it is clear that there is a tradeoff between stepsize and accuracy. In practice, we suggest to conduct some pilot numerical analysis to identify a proper step size.

4.5 Computational Efficiency

We report the computational time of our proposed methods as functions of p, q and n separately. The data are simulated from Model III with SNR = 0.25, $r^* = 3$ and $\rho = 0.3$. In each setup, we let one of the three model dimensions, i.e., the sample size n, the number of predictors p, and the number of responses q, to vary from 200 to 2000, while holding the other two at a constant value of 200. The experiment is repeated 100 times under each setting. Figure 5 reports the average computation times as functions of p, q and n in three panels from the left to the right, respectively. It is evident that our proposed approaches are scalable to large-scale problems, and the gain over the ACS-based approaches is dramatic.

Table 1: Results for Model II with SNR = 0.5, $p \in \{100, 200, 400\}$ and $r^* \in \{3, 6\}$; here Er(C) and Er(XC) are rescaled by multiplying 10^3.

| Method	Er(C)	Er(XC)	FPR (%)	FNR (%)	Time (s)	Er(C)	Er(XC)	FPR (%)	FNR (%)	Time (s)
			$p = 100$					$p = 100$		
RRR	4478.89	246.22	100.00	0.00	0.08	4788.37	478.79	100.00	0.00	0.09
SRRR	1.66	139.64	57.32	0.00	5.28	4.09	302.34	56.12	0.00	7.26
SOFAR	6.44	202.75	24.59	4.83	10.70	29.85	855.91	22.83	7.75	22.13
SeqACS	1.34	85.83	0.96	4.54	0.50	4.83	289.24	2.45	9.01	1.56
ParACS(L)	0.97	70.50	2.10	3.56	8.08	6.02	402.34	4.82	6.50	8.55
ParACS(R)	1.54	92.22	3.24	5.92	7.00	4.71	269.37	8.38	6.11	7.45
SeqSTL	0.65	42.58	0.85	4.52	0.38	3.76	215.73	2.37	10.90	0.76
ParSTL(L)	0.70	52.01	1.12	3.94	1.07	5.83	380.29	3.19	8.87	1.17
ParSTL(R)	0.75	46.16	1.06	3.96	0.15	3.57	188.62	3.17	8.03	0.13
			$p = 200$					$p = 200$		
RRR	18.99	247.89	100.00	0.00	0.12	91.45	472.36	100.00	0.00	0.11
SRRR	0.85	160.50	64.31	0.00	6.98	1.70	287.74	65.43	0.00	14.38
SOFAR	3.45	142.55	29.23	3.87	13.51	20.80	744.53	21.12	7.25	25.80
SeqACS	0.78	99.79	0.67	4.46	1.68	2.87	340.08	1.56	10.45	5.17
ParACS(L)	0.55	80.38	1.58	3.33	8.84	2.42	316.90	3.58	6.79	9.50
ParACS(R)	0.77	95.32	2.03	8.71	8.08	2.25	273.33	4.51	8.90	8.45
SeqSTL	0.41	50.72	0.59	4.12	0.47	2.25	250.90	1.53	12.55	1.01
ParSTL(L)	0.42	61.85	0.79	3.77	1.03	2.45	306.59	2.14	9.33	1.11
ParSTL(R)	0.46	54.22	0.75	3.85	0.17	2.46	244.62	2.18	8.96	0.18
			$p = 400$					$p = 400$		
RRR	13.86	243.25	100.00	0.00	0.47	66.92	478.95	100.00	0.00	0.43
SRRR	0.49	194.07	50.02	0.00	21.94	0.99	355.65	57.26	0.00	40.66
SOFAR	5.61	158.58	32.99	3.27	30.08	29.44	554.55	26.58	7.45	10.38
SeqACS	0.46	115.37	0.44	5.17	12.47	1.80	404.84	0.92	10.43	30.04
ParACS(L)	0.31	87.26	1.13	4.12	12.41	1.22	305.48	2.42	7.29	12.99
ParACS(R)	0.46	106.39	1.22	6.40	12.83	1.43	306.86	2.68	9.18	14.19
SeqSTL	0.24	57.48	0.37	4.87	1.03	1.42	281.50	0.94	12.76	1.77
ParSTL(L)	0.24	67.41	0.50	4.48	1.41	1.57	349.37	1.36	9.97	1.66
ParSTL(R)	0.26	61.04	0.45	4.67	0.51	1.56	278.45	1.36	9.49	0.54

(a) SNR = 0.5, $r = 3$	(b) SNR = 0.5, $r = 6$

Table 2: Results for Model III with SNR $= 0.5$, $p \in \{100, 200, 400\}$ and $r^* \in \{3, 6\}$; here Er(C) and Er(XC) are rescaled by multiplying 10^3.

Method	Er(C)	Er(XC)	FPR (%)	FNR (%)	Time (s)	Er(C)	Er(XC)	FPR (%)	FNR (%)	Time (s)
			$p = 100$					$p = 100$		
RRR	5628.05	241.69	100.00	0.00	0.08	10574.61	472.34	100.00	0.00	0.08
SRRR	1.86	148.12	62.70	0.00	4.82	5.76	365.04	66.86	0.00	6.63
SOFAR	3.88	150.65	23.48	0.00	11.27	12.38	538.21	16.28	0.25	20.74
SeqACS	1.15	76.76	1.07	0.31	0.38	5.13	334.67	2.60	1.85	0.88
ParACS(L)	1.17	73.86	4.12	0.00	8.11	5.36	331.75	12.05	0.27	8.61
ParACS(R)	2.05	115.89	9.37	0.64	7.05	7.77	380.04	23.52	2.74	7.64
SeqSTL	0.64	43.43	1.23	0.57	0.35	3.87	246.22	3.42	17.39	0.88
ParSTL(L)	0.94	61.82	2.65	0.26	1.13	5.33	336.18	8.73	5.58	1.15
ParSTL(R)	0.90	57.82	3.22	1.12	0.15	4.34	230.55	10.15	4.70	0.13
			$p = 200$					$p = 200$		
RRR	18.05	246.27	100.00	0.00	0.12	83.35	479.98	100.00	0.00	0.11
SRRR	0.98	166.66	66.31	0.00	6.54	2.42	328.58	71.09	0.00	10.14
SOFAR	1.82	114.63	27.10	0.00	12.91	5.92	401.01	15.58	0.00	27.03
SeqACS	0.62	83.42	0.73	0.62	1.15	2.87	365.76	1.72	2.81	2.64
ParACS(L)	0.64	80.90	2.87	0.00	9.02	2.96	317.90	8.30	0.06	9.25
ParACS(R)	0.62	75.61	2.81	0.00	8.25	2.23	224.24	8.33	0.00	8.37
SeqSTL	0.36	47.35	0.67	0.67	0.43	2.09	258.97	2.09	17.87	1.12
ParSTL(L)	0.53	66.74	1.71	1.10	0.94	3.00	319.34	5.64	7.87	1.17
ParSTL(R)	0.56	67.50	2.09	0.76	0.17	2.85	272.97	6.75	5.62	0.20
			$p = 400$					$p = 400$		
RRR	13.53	247.87	100.00	0.00	0.46	64.31	478.36	100.00	0.00	0.45
SRRR	0.58	205.47	51.19	0.00	20.32	1.48	384.30	59.62	0.00	31.72
SOFAR	4.09	158.56	31.15	0.26	31.04	15.87	504.32	22.85	5.14	10.49
SeqACS	0.35	95.23	0.40	0.33	7.20	1.54	396.73	1.01	2.49	19.71
ParACS(L)	0.37	94.57	1.73	0.00	12.67	1.73	337.57	5.16	0.05	12.67
ParACS(R)	0.37	89.29	1.89	0.00	12.65	1.54	270.46	6.17	0.00	13.25
SeqSTL	0.20	52.19	0.36	0.67	0.96	1.12	276.18	1.12	15.96	1.80
ParSTL(L)	0.29	75.08	0.92	0.50	1.33	1.84	358.94	3.25	8.52	1.66
ParSTL(R)	0.32	75.86	1.16	0.81	0.53	1.87	316.16	4.02	6.83	0.59

(a) SNR $= 0.5$, $r = 3$ (b) SNR $= 0.5$, $r = 6$
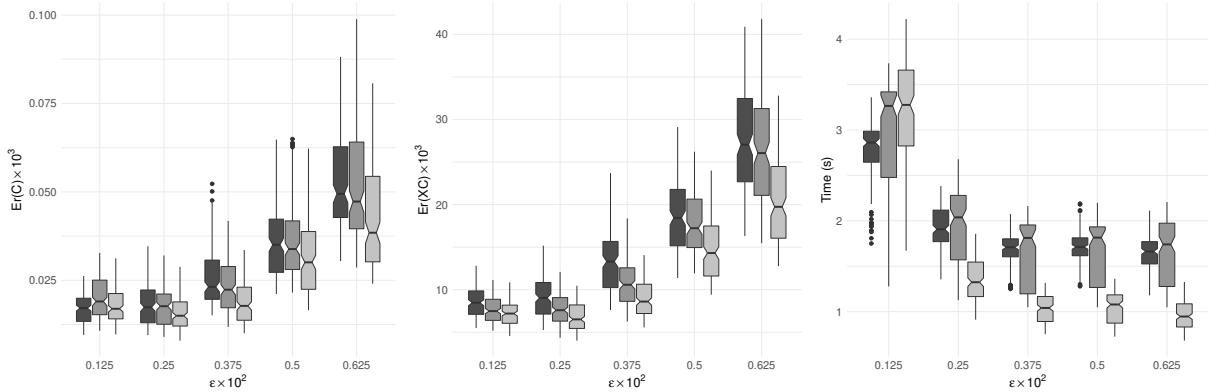
Figure 4: Impact of the step size ϵ on the performance of the stagewise learning methods. The three colors, dark, medium and light grey correspond to ParSTL(L), ParSTL(R), and SeqSTL respectively.
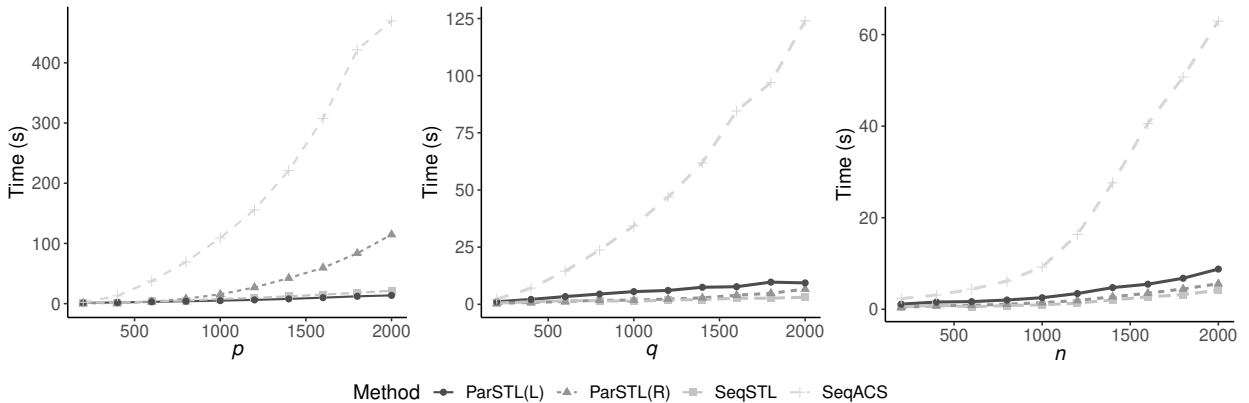
Figure 5: Simulation: Computation cost with increasing p, q and n.

5. Yeast eQTL Mapping Analysis

In an expression quantitative trait loci (eQTLs) mapping analysis, the main objective is to examine the association between the eQTLs, i.e., regions of the genome containing DNA sequence variants, and the expression levels of the genes in certain signaling pathways. Biochemical evidence often suggests that there exist a few functionally distinct signaling pathways of genes, each of which may involve only a subset of genes and correspondingly a subset of eQTLs. Therefore, the recovery of such association structure can be formulated as a sparse factor regression problem, with the gene expressions being the responses and the eQTLs being the predictors. Here, we analyze the yeast eQTL data set described by Brem and Kruglyak (2005) and Storey et al. (2005), to illustrate the power and scalability of the proposed approaches for estimating the associations between $p = 3244$ genetic markers and $q = 54$ genes that belong to the yeast Mitogen-activated protein kinases (MAPKs) signaling pathway (Kanehisa et al., 2009), with data collected from $n = 112$ yeast samples.

Uematsu et al. (2019) used the same data set to showcase their SOFAR method. In their work, a marginal screening approach was first used to reduce the number of marker locations from $p = 3244$ to $p = 605$, which greatly alleviated the computational burden. However, since our proposed approaches are more scalable, it is worth trying to include all the markers for a joint regression. We thus try both approaches in this analysis.

Under either setting, i.e., with or without marginal screening, we perform a random splitting procedure to compare the performance of different methods. To make the comparison fair, all the methods have the same pre-specified rank, which is selected from RRR via 10-fold cross validation. Specifically, each time the data set is randomly split into 80% for model fitting and 20% for computing the out-sample mean squared error (MSE) of the fitted model. Also recorded are the computation time (in seconds), the number of nonzero entries in $\widehat{\mathbf{U}}$ ($\|\widehat{\mathbf{U}}\|_0$), the number of nonzero rows in $\widehat{\mathbf{U}}$ ($\|\widehat{\mathbf{U}}\|_{2,0}$), the number of nonzero entries in $\widehat{\mathbf{V}}$ ($\|\widehat{\mathbf{V}}\|_0$), and the number of nonzero rows in $\widehat{\mathbf{V}}$ ($\|\widehat{\mathbf{V}}\|_{2,0}$). The procedure is repeated 100 times, and to make a robust comparison we compute the 10% trimmed means and standard deviations of the above performance measures.

Table 3 reports the results for the setting with marginal screening. All the methods that pursue sparse and low rank structures outperform the benchmark RRR in term of out-of-sample prediction performance; the ParSTL(R) method performs the best, although the improvement

Table 3: Yeast eQTL mapping analysis: Results with marginal screening.

Method	$\|\widehat{\mathbf{U}}\|_0$	$\|\widehat{\mathbf{U}}\|_{2,0}$	$\|\widehat{\mathbf{V}}\|_0$	$\|\widehat{\mathbf{V}}\|_{2,0}$	MSE	Time(s)
RRR	1815 (0)	605 (0)	162 (0)	54 (0)	0.34 (0.03)	0.52 (0.11)
SOFAR	128.09 (23.85)	44.26 (7.64)	43.5 (7.38)	14.96 (1.41)	0.26 (0.03)	85.35 (18.55)
RSSVD	61.39 (10.94)	58.49 (10.52)	6.74 (2.13)	6.2 (1.16)	0.3 (0.03)	38.14 (13.93)
SRRR	522.45 (18.28)	174.15 (6.09)	162 (0)	54 (0)	0.24 (0.02)	76.67 (13.78)
SeqACS	38.61 (3.34)	38.6 (3.35)	14.51 (1.44)	11.11 (0.93)	0.26 (0.02)	29.84 (10.16)
SeqSTL	67.42 (8.03)	64.49 (7.52)	31.64 (4.28)	19.06 (1.9)	0.23 (0.02)	3.26 (0.82)
ParSTL(L)	59.92 (9.39)	45.85 (6.21)	20.43 (2.61)	13.5 (1.66)	0.29 (0.1)	1.34 (0.33)
ParSTL(R)	72.71 (8.84)	69.62 (8.17)	34.50 (4.68)	21.25 (1.92)	0.21 (0.02)	1.12 (0.21)

Table 4: Yeast eQTL mapping analysis: Results with full data (without marginal screening).

Method	$\|\widehat{\mathbf{U}}\|_0$	$\|\widehat{\mathbf{U}}\|_{2,0}$	$\|\widehat{\mathbf{V}}\|_0$	$\|\widehat{\mathbf{V}}\|_{2,0}$	MSE	Time(s)
RRR	9732 (1788.02)	3244 (0)	162 (29.76)	54 (0)	0.44 (0.04)	32.35 (2.60)
SOFAR	156.62 (84.95)	61.1 (25.4)	40.09 (13.07)	16.02 (2.54)	0.28 (0.03)	283.51 (102.01)
RSSVD	52 (9.4)	51.09 (9.17)	5.4 (0.7)	5.35 (0.62)	0.3 (0.02)	1330.67 (541.91)
SRRR	1332.09 (359.29)	437.06 (44.94)	162 (29.76)	54 (0)	0.24 (0.02)	900.95 (248.2)
SeqACS	38.74 (3.7)	38.67 (3.74)	13.29 (1.35)	10.31 (0.91)	0.26 (0.02)	1378.98 (415.3)
SeqSTL	57.56 (9.09)	56.46 (8.5)	24.18 (5.37)	15.65 (2.37)	0.24 (0.02)	16 (5.04)
ParSTL(L)	57.69 (12.03)	43.67 (5.58)	19.21 (3.34)	12.36 (1.5)	0.31 (0.12)	5.41 (1.7)
ParSTL(R)	67.46 (13.01)	64 (8.35)	30.91 (7.85)	19.25 (2.7)	0.22 (0.02)	36.02 (3.16)

is not substantial comparing to other close competitors. Our proposed methods are no doubt the most computationally efficient among all the sparse and low-rank methods. The gain in computational efficiency is even more revealing and dramatic in the setting of $p = 3244$ without marginal screening, for which the results are reported in Table 4. For example, ParSTL can be more than 100 times faster than competitors such as RSSVD (similar to ParACS(R)) or SeqACS. It is also interesting to see that the predictive performance of all methods becomes slightly worse comparing to the setting with marginal screening. The performance of RRR deteriorates the most since it lacks the power of eliminating noise variables. The results suggest that in this particular application the potential benefit of jointly considering all the markers is exceeded by the loss due to noise accumulation. Nevertheless, we see that the proposed methods are still able to perform competitively in such a high-dimensional problem with excellent scalability.

Lastly, we examine the genes selected in the estimated pathways (low-rank components or layers) using our proposed methods with the full data set. Since each $\widehat{\mathbf{v}}_k$ vector is normalized to have unit ℓ_1 norm, we define top genes to be those with entries larger than $1/q$ in magnitude. The selected genes are summarized in Table 5, in which the results in Uematsu et al. (2019) are reproduced for each of comparison. Figure 6 shows the scatterplots of the latent responses $\mathbf{Y}\widehat{\mathbf{v}}_k$ versus the latent predictors $\mathbf{X}\widehat{\mathbf{u}}_k$ from the three stagewise methods fitted on the full data. The results from the three proposed methods are mostly consistent with Uematsu et al. (2019). The patterns recovered by the three methods are similar, except that ParSTL(L) appears to reveal a slight different association comparing to the other two methods. As explained in Gustin et al. (1998), pheromone induces mating and nitrogen starvation induces filamentation are two pathways in yeast cells. The identified genes STE2, STE3 and GPA1 are receptors required for mating pheromone, which bind the cognate lipopeptide pheromones (MFA2, MFA1 etc). Besides,

Table 5: Yeast eQTL mapping analysis: Top genes in the estimated pathways. The first row reproduces the results of Uematsu et al. (2019). The first panel is for the setting with marginal screening, and the second panel is for the setting with full data.

Method	Layer 1	Layer 2	Layer 3
With marginal screening ($p = 605$)			
SOFAR	STE3, STE2, MFA2, MFA1	CTT1, SLN1, SLT2, MSN4, GLO1	FUS1, FAR1, STE2, STE3, GPA1, FUS3, STE12
SeqSTL	STE3, STE2, MFA2, MFA1, CTT1, FUS1	CTT1, FUS1, MSN4, GLO1, SLN1	FUS1, FAR1, STE2, GPA1, FUS3, STE3, TEC1, SLN1, STE12, MFA2, CTT1
ParSTL(R)	STE3, STE2, MFA2, MFA1, CTT1, FUS1, TEC1	CTT1, FUS1, MSN4, STE3, GLO1, SLN1, MFA2	FUS1, FAR1, STE2, GPA1, FUS3, STE3, TEC1, CTT1, SLN1, STE12
ParSTL(L)	STE3, STE2, MFA2, MFA1, CTT1, FUS1	CTT1, FUS1, MSN4, GLO1	CTT1, MSN4, STE2
Full data ($p = 3244$)			
SeqSTL	STE3, STE2, MFA2, MFA1, CTT1, FUS1	FUS1, CTT1, FAR1, SLN1, STE2, MFA2, GPA1	TEC1
ParSTL(R)	STE3, STE2, MFA2, MFA1, CTT1, FUS1, TEC1, WSC2	CTT1, FUS1, MSN4, GLO1, MFA2, STE3, SLN1	FUS1, FAR1, STE2, STE3, GPA1, FUS3, TEC1, CTT1
ParSTL(L)	STE3, STE2, MFA2, MFA1, CTT1, FUS1	CTT1, FUS1, MSN4	CTT1, MSN4, STE2

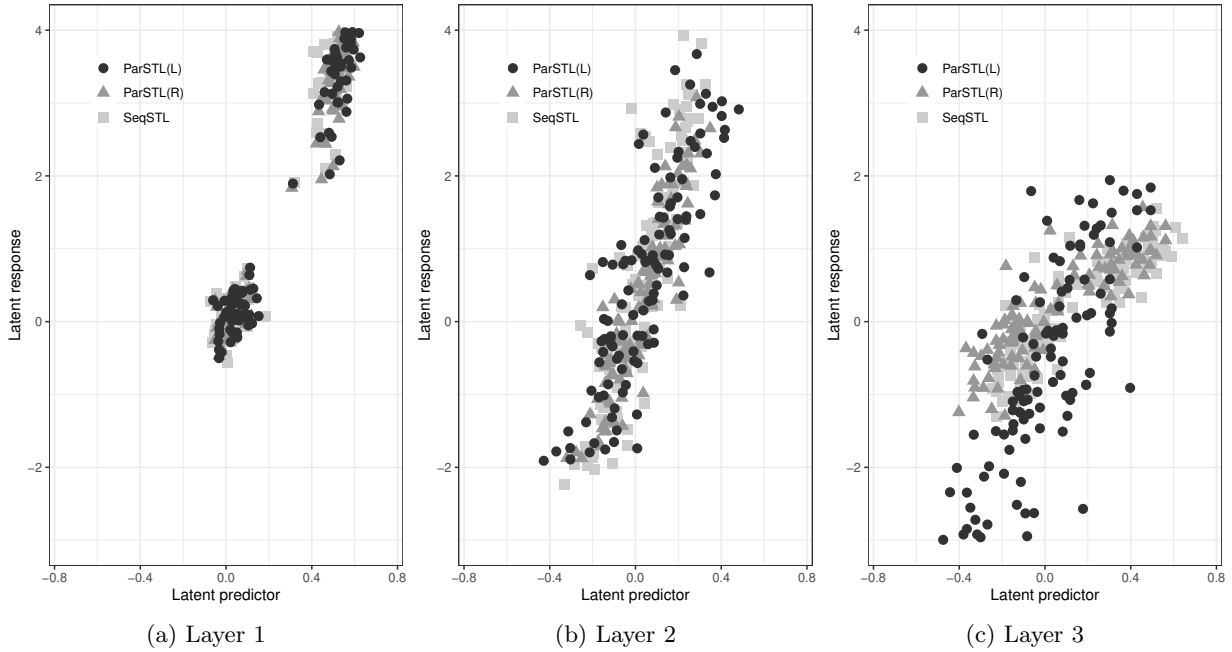(a) Layer 1   (b) Layer 2   (c) Layer 3

Figure 6: Yeast eQTL mapping analysis: Scatter plots of the estimated latent responses and latent predictors.

another top gene FUS3 is used to phosphorylate several downstream targets, including FAR1 which mediates various responses required for successful mating. Interestingly, our analysis identified an additional gene, TEC1, that was not reported in previous analysis. TEC1, as the transcription factor specific to the filamentation pathway, has been shown to have FUS3-dependent degradation induced by pheromone signaling (Bao et al., 2004).

6. Discussion

We have proposed a statistically guided stagewise learning approach for sparse factorization of large matrix. Both sequential and parallel deflation strategies are analyzed with corresponding statistical underpinnings. Moreover, a contended stagewise learning technique is developed to efficiently trace out the whole solution paths, which enjoys a much lower computational complexity than the alternating convex search. Our work is among the first to enable stagewise learning for non-convex problems, and extensive numerical studies demonstrate the effectiveness and scalability of our approach. There are several future research directions. Our stagewise learning algorithm applies to the CURE problem where the penalty on the unit-rank coefficient matrix is in the form of the elastic net (Zou and Hastie, 2005); although the ℓ_2 term is considered as fixed in this work, tuning it may lead to improved predictive performance. It is also interesting to explore other shrinkage and sparsity-inducing penalty forms in the CURE problem. Our approach can be applicable in many multi-convex problems such as multi-modal data and tensor data analysis, and it can also be extended to more general model settings such as generalized linear models and mixture models. Building upon the architecture of the contended stagewise learning procedure, it is hopeful that a general framework of non-convex stagewise learning can be developed. Besides deflation methods, we will also explore other scalable alternatives for the sparse factor regression problem. In particular, a "lazy sparse reduced rank regression" strategy as described in Qian et al. (2022) could be promising, where the first step obtains an initial estimator of $\mathbf{C}^*$, and the second step takes the initial estimator as the input to perform the sparse SVD. To make it work in our setting, we may extend the existing fast sparse SVD solvers (Yang et al., 2014) to the generalized $\mathbf{P}$-orthogonal SVD setup.

Acknowledgment

Appendix

We present the proofs of the theoretical results in this section. For matrices $\mathbf{A}$ and $\mathbf{B}$, $\langle \mathbf{A}, \mathbf{B} \rangle$ means the Frobenius inner product of $\mathbf{A}$ and $\mathbf{B}$, $\|\mathbf{A}\|_F$ is the Forbenius norm of $\mathbf{A}$, and $\|\mathbf{A}\|_1 = \sum_{i,j} |A_{ij}|$. Moreover, denote by J_k the support set of $\mathbf{C}_k^*$ and $\widehat{\boldsymbol{\Delta}}_k = \widehat{\mathbf{C}}_k - \mathbf{C}_k^*$ with $\widehat{\mathbf{C}}_k$ the kth layer estimator of $\mathbf{C}_k^*$ in either sequential or parallel pursuit.

A. Proof for the sequential pursuit

Recall that the sequential pursuit with lasso penalty sequentially solves

$$
(\widehat{d}_k, \widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k) = \arg\min_{(d, \mathbf{u}, \mathbf{v})} (2n)^{-1} \|\mathbf{Y}_k - d\mathbf{X}\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_F^2 + \lambda_k \|d\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_1
$$
$$
\text{s.t. } d > 0, n^{-1}\mathbf{u}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{u} = \mathbf{v}^{\mathrm{T}}\mathbf{v} = 1, \tag{16}
$$

where $\mathbf{Y}_k = \mathbf{Y} - \mathbf{X}\sum_{\ell=0}^{k-1} \widehat{\mathbf{C}}_\ell$, $\widehat{\mathbf{C}}_0 = \mathbf{0}$ and $\widehat{\mathbf{C}}_\ell = \widehat{d}_\ell \widehat{\mathbf{u}}_\ell \widehat{\mathbf{v}}_\ell^{\mathrm{T}}$. We need two lemmas before showing the main theorem.

A.1 Lemma 1 and its proof

Lemma 1 *Under Condition 2.2, for any k, $1 \le k \le r^* - 1$, it holds that*

$$
\frac{\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2}{2|\sum_{j=k+1}^{r^*} \langle \mathbf{X}\mathbf{C}_j^*, \mathbf{X}\widehat{\mathbf{C}}_k \rangle|} \ge \omega_k^{-1} > 1,
$$

where $\omega_k = 1 - \delta_k^*/d_k^* > 0$.

Proof [Proof of Lemma 1] Since the following argument applies to any k, $1 \le k \le r^* - 1$, we consider a fixed k here. First of all, there exists the following decomposition for $\mathbf{X}\widehat{\mathbf{u}}_k$ and $\widehat{\mathbf{v}}_k$ that

$$
\frac{1}{\sqrt{n}}\mathbf{X}\widehat{\mathbf{u}}_k = \sum_{j=1}^{r^*+1} a_j \frac{1}{\sqrt{n}}\mathbf{X}\mathbf{u}_j^*, \quad \widehat{\mathbf{v}}_k = \sum_{j=1}^{r^*+1} b_j \mathbf{v}_j^*, \tag{17}
$$

where $\frac{1}{\sqrt{n}}\mathbf{X}\mathbf{u}_{r^*+1}^*$ and $\mathbf{v}_{r^*+1}^*$ are some unit length vectors orthogonal to the sets of unit length orthogonal vectors $\{\frac{1}{\sqrt{n}}\mathbf{X}\mathbf{u}_j^*\}_{j=1}^{r^*}$ and $\{\mathbf{v}_j^*\}_{j=1}^{r^*}$, respectively, so that the coefficients satisfy $\sum_{j=1}^{r^*+1} a_j^2 = \sum_{j=1}^{r^*+1} b_j^2 = 1$.

If $a_k = 1$ or $b_k = 1$, it is a relatively trivial case. So we consider cases where $a_k < 1$ and $b_k < 1$. Based on decomposition (17), direct calculation yields that

$$
n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2 = \widehat{d}_k^2 - 2\widehat{d}_k d_k^* a_k b_k + d_k^{*2},
$$
$$
n^{-1}\sum_{j=k+1}^{r^*} \langle \mathbf{X}\mathbf{C}_j^*, \mathbf{X}\widehat{\mathbf{C}}_k \rangle = \sum_{j=k+1}^{r^*} d_j^* \widehat{d}_k a_j b_j.
$$

It gives

$$\frac{\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2}{2|\sum_{j=k+1}^{r^*} \langle \mathbf{X}\mathbf{C}_j^*, \mathbf{X}\widehat{\mathbf{C}}_k\rangle|} = \frac{\widehat{\tau}_k^2 - 2\widehat{\tau}_k a_k b_k + 1}{2\widehat{\tau}_k |\sum_{j=k+1}^{r^*} (d_j^*/d_k^*)a_j b_j|}$$

$$= \frac{\widehat{\tau}_k - 2a_k b_k + \widehat{\tau}_k^{-1}}{2|\sum_{j=k+1}^{r^*} (d_j^*/d_k^*)a_j b_j|} \geq \frac{1 - a_k b_k}{|\sum_{j=k+1}^{r^*} (d_j^*/d_k^*)a_j b_j|},$$

where $\widehat{\tau}_k = \widehat{d}_k/d_k^* > 0$ and we make use of the fact that $\widehat{\tau}_k + \widehat{\tau}_k^{-1} \geq 2$.

Then by Condition 2.2, we have

$$\max_{k+1 \leq j \leq r^*} d_j^*/d_k^* = d_{k+1}^*/d_k^* = (d_k^* - \delta_k^*)/d_k^* = 1 - \delta_k^*/d_k^* = \omega_k < 1.$$

Denote by $\widetilde{\omega}_j = d_j^*/d_k^*$. It follows that

$$\frac{(1 - a_k b_k)^2}{(\sum_{j=k+1}^{r^*} \widetilde{\omega}_j a_j b_j)^2} \geq \frac{(1 - a_k b_k)^2}{(\sum_{j=k+1}^{r^*} \widetilde{\omega}_j a_j^2)(\sum_{j=k+1}^{r^*} \widetilde{\omega}_j b_j^2)}$$

$$\geq \frac{(1 - a_k b_k)^2}{\omega_k^2(\sum_{j=k+1}^{r^*} a_j^2)(\sum_{j=k+1}^{r^*} b_j^2)} \geq \frac{(1 - a_k b_k)^2}{\omega_k^2(1 - a_k^2)(1 - b_k^2)} \geq \omega_k^{-2}.$$

Thus, it yields

$$\frac{\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2}{2|\sum_{\ell=k+1}^{r^*} \langle \mathbf{X}\mathbf{C}_\ell^*, \mathbf{X}\widehat{\mathbf{C}}_k\rangle|} \geq \frac{1 - a_k b_k}{|\sum_{j=k+1}^{r^*} (d_j^*/d_k^*)a_j b_j|} \geq \omega_k^{-1} > 1,$$

which concludes the results of Lemma 1. ∎

A.2 Lemma 2 and its proof

Lemma 2 *Under Conditions 2.1–2.3 with $s \geq s_1$ and $\lambda_1 = 2\sigma_{\max}\sqrt{2\alpha \log(pq)/n}$ for some constant $\alpha > 1$, we have with probability at least $1 - (pq)^{1-\alpha}$,*

$$\|\widehat{\boldsymbol{\Delta}}_1\|_F = O(\sqrt{s_1}\lambda_1) = O\left(\theta_1\sqrt{s_1 \log(pq)/n}\right),$$

$$\|\widehat{\boldsymbol{\Delta}}_1\|_1 = O(s_1\lambda_1) = O\left(\theta_1 s_1\sqrt{\log(pq)/n}\right),$$

where $\theta_1 = (1 - \omega_1)^{-1} = d_1^*/\delta_1^*$.

Proof [Proof of Lemma 2] Because $(\widehat{d}_1, \widehat{\mathbf{u}}_1, \widehat{\mathbf{v}}_1)$ is the optimal solution of (16) for $k = 1$, we have

$$(2n)^{-1}\|\mathbf{Y} - \mathbf{X}\widehat{\mathbf{C}}_1\|_F^2 + \lambda_1\|\widehat{\mathbf{C}}_1\| \leq (2n)^{-1}\|\mathbf{Y} - \mathbf{X}\mathbf{C}_1^*\|_F^2 + \lambda_1\|\mathbf{C}_1^*\|_1,$$

where $\widehat{\mathbf{C}}_1 = \widehat{d}_1\widehat{\mathbf{u}}_1\widehat{\mathbf{v}}_1^T$ and $\mathbf{C}_1^* = d_1^*\mathbf{u}_1^*\mathbf{v}_1^{*T}$. After some simplification, the above inequality gives that

$$(2n)^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_1\|_F^2 + \lambda_1\|\widehat{\mathbf{C}}_1\|_1 \leq n^{-1}\langle \mathbf{X}^T\mathbf{E}, \widehat{\boldsymbol{\Delta}}_1\rangle + n^{-1}\sum_{k=2}^{r^*} \langle \mathbf{X}\mathbf{C}_k^*, \mathbf{X}\widehat{\mathbf{C}}_1\rangle + \lambda_1\|\mathbf{C}_1^*\|_1.$$

Applying Lemma 1, we get

$$(2n)^{-1}(1-\omega_1)\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_1\|_F^2 + \lambda_1\|\widehat{\mathbf{C}}_1\|_1 \le n^{-1}\langle\mathbf{X}^{\mathrm{T}}\mathbf{E}, \widehat{\boldsymbol{\Delta}}_1\rangle + \lambda_1\|\mathbf{C}_1^*\|_1$$
$$\le n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{E}\|_{\max}\|\widehat{\boldsymbol{\Delta}}\|_1 + \lambda_1\|\mathbf{C}_1^*\|_1, \tag{18}$$

where the last inequality holds because of the Hölder's inequality. Then on event $\mathcal{A} = \{n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{E}\|_{\max} \le \lambda_1/2\}$, by inequality (18), we have

$$\lambda_1\|\widehat{\mathbf{C}}_1\|_1 \le \frac{\lambda_1}{2}\|\widehat{\boldsymbol{\Delta}}_1\|_1 + \lambda_1\|\mathbf{C}_1^*\|_1.$$

It yields that

$$\|\widehat{\boldsymbol{\Delta}}_{J_1^c}\|_1 \le 3\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_1. \tag{19}$$

Therefore, by Condition 2.3, we can get

$$\rho_l(\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_F^2 \vee \|\widehat{\boldsymbol{\Delta}}_{J_{1,s_1}^c}\|_F^2) \le n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_1\|_F^2, \tag{20}$$

where $\widehat{\boldsymbol{\Delta}}_{J_{1,s_1}^c}$ is the submatrix of $\widehat{\boldsymbol{\Delta}}_{J_1^c}$ as defined in Condition 2.3. Our discussion will be conditioning on the event $\mathcal{A}$ hereafter.

On the other hand, combining inequalities (18) and (19) yields that

$$(1-\omega_1)(2n)^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_1\|_F^2 \le (3\lambda_1/2)\|\widehat{\boldsymbol{\Delta}}_1\|_1 \le 6\lambda_1\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_1.$$

Together with (20), we have

$$\frac{1-\omega_1}{2}\rho_l(\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_F^2 \vee \|\widehat{\boldsymbol{\Delta}}_{J_{1,s_1}^c}\|_F^2) \le (1-\omega_1)(2n)^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_1\|_F^2 \le 6\lambda_1\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_1 \le 6\lambda_1\sqrt{s_1}\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_F. \tag{21}$$

It follows that

$$\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_F \le 12(\rho_l - \rho_l\omega_1)^{-1}\lambda_1\sqrt{s_1}. \tag{22}$$

Note that the kth largest component of $\widehat{\boldsymbol{\Delta}}_{J_1^c}$ in terms of absolute value is bounded from above by $\|\widehat{\boldsymbol{\Delta}}_{J_1^c}\|_1/k$, then we have

$$\|\widehat{\boldsymbol{\Delta}}_{\tilde{J}_{1,s_1}^c}\|_F^2 \le \sum_{k=s_1+1}^{pq-s_1}\|\widehat{\boldsymbol{\Delta}}_{J_1^c}\|_1^2/k^2 \le s_1^{-1}\|\widehat{\boldsymbol{\Delta}}_{J_1^c}\|_1^2,$$

where $\widehat{\boldsymbol{\Delta}}_{\tilde{J}_{1,s_1}^c}$ is a submatrix of $\widehat{\boldsymbol{\Delta}}_{J_1^c}$ consisting of the components excluding those with the s_1 largest absolute values. Then by inequalities (19) and (22), we get

$$\|\widehat{\boldsymbol{\Delta}}_{\tilde{J}_{1,s_1}^c}\|_F \le s_1^{-1/2}\|\widehat{\boldsymbol{\Delta}}_{J_1^c}\|_1 \le 3s_1^{-1/2}\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_1 \le 3\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_F \le 36(\rho_l - \rho_l\omega_1)^{-1}\lambda_1\sqrt{s_1}. \tag{23}$$

In view of inequalities (21) and (22), we can also get

$$\|\widehat{\boldsymbol{\Delta}}_{J_{1,s_1}^c}\|_F \le \{12\lambda_1(\rho_l - \rho_l\omega_1)^{-1}\sqrt{s_1}\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_F\}^{1/2} \le 12(\rho_l - \rho_l\omega_1)^{-1}\lambda_1\sqrt{s_1}. \tag{24}$$

Finally, combining (22), (23) and (24), we have

$$\|\widehat{\boldsymbol{\Delta}}_1\|_F \le \|\widehat{\boldsymbol{\Delta}}_{J_1}\|_F + \|\widehat{\boldsymbol{\Delta}}_{\tilde{J}_{1,s_1}^c}\|_F + \|\widehat{\boldsymbol{\Delta}}_{J_{1,s_1}^c}\|_F \le 5M_1\sqrt{s_1}\lambda_1 = O(\theta_1\sqrt{s_1}\lambda_1) = O\left(\theta_1\sqrt{s_1\log(pq)/n}\right),$$

where $M_1 = 12(\rho_l - \rho_l\omega_1)^{-1}$ and $\theta_1 = (1 - \omega_1)^{-1}$. For ℓ_1 loss bound, from inequalities (19) and (22), we directly get

$$\|\widehat{\boldsymbol{\Delta}}_1\|_1 \leq 4\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_1 \leq 4\sqrt{s_1}\|\widehat{\boldsymbol{\Delta}}_{J_1}\|_F \leq 4M_1 s_1 \lambda_1 = O(\theta_1 s_1 \lambda_1) = O\left(\theta_1 s_1 \sqrt{\log(pq)/n}\right).$$

At last, we will derive the probability of event $\mathcal{A}$. First, it follows from the union bound that

$$P\{n^{-1}\|\mathbf{X}^T\mathbf{E}\|_{\max} > \lambda_1/2\} \leq \sum_{i=1}^{p}\sum_{j=1}^{q} P\{n^{-1}|\mathbf{x}_i^T\mathbf{e}_j| > \lambda_1/2\}.$$

Since $\mathrm{Var}(\mathbf{x}_i^T\mathbf{e}_j) = \sigma_j^2\|\mathbf{x}_i\|_2^2 \leq n\sigma_{\max}^2$, under Condition 2.1, applying the tail probability bound of Gaussian distribution, we get

$$P\{n^{-1}\|\mathbf{X}^T\mathbf{E}\|_{\max} > \lambda_1/2\} \leq 2pq\exp\{-\frac{n\lambda_1^2}{8\sigma_{\max}^2}\}. \tag{25}$$

Therefore, when $\lambda_1^2 = 8\alpha\sigma_{\max}^2 n^{-1}\log(pq)$, we know that event $\mathcal{A}$ holds with probability at least $1 - 2(pq)^{1-\alpha}$. It completes the proof of Lemma 2. ∎

A.3 Proof of Theorem 1

When $k = 1$, the results of Theorem 1 are established in Lemma 2. Now we do some preparation before showing the results for $k > 1$. Note that

$$\|\mathbf{Y}_k - d\mathbf{X}\mathbf{u}\mathbf{v}^T\|_F^2 = \|\mathbf{Y} - \mathbf{X}\sum_{\ell=0}^{k-1}\widehat{\mathbf{C}}_\ell - d\mathbf{X}\mathbf{u}\mathbf{v}^T\|_F^2 = \|\widetilde{\mathbf{Y}}_k - \sum_{\ell=0}^{k-1}\mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell - d\mathbf{X}\mathbf{u}\mathbf{v}^T\|_F^2$$

$$= \|\widetilde{\mathbf{Y}}_k - d\mathbf{X}\mathbf{u}\mathbf{v}^T\|_F^2 + 2\sum_{\ell=0}^{k-1}\langle d\mathbf{X}\mathbf{u}\mathbf{v}^T, \mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell\rangle + T,$$

where $\widetilde{\mathbf{Y}}_k = \mathbf{X}\sum_{\ell=k}^{r^*}\mathbf{C}_\ell + \mathbf{E}$, $\widetilde{\boldsymbol{\Delta}}_\ell = \widehat{\mathbf{C}}_\ell - \mathbf{C}_\ell^*$ and $T = -2\sum_{\ell=0}^{k-1}\langle\mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell, \widetilde{\mathbf{Y}}_k\rangle$. Because T doesn't change with the triple $(d, \mathbf{u}, \mathbf{v})$, the optimization problem (16) is equivalent to

$$(\widehat{d}_k, \widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k) = \arg\min_{d,\mathbf{u},\mathbf{v}} (2n)^{-1}\|\widetilde{\mathbf{Y}}_k - d\mathbf{X}\mathbf{u}\mathbf{v}^T\|_F^2 + n^{-1}\sum_{\ell=0}^{k-1}\langle d\mathbf{X}\mathbf{u}\mathbf{v}^T, \mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell\rangle$$
$$+ \lambda_k\|d\mathbf{u}\mathbf{v}^T\|_1. \tag{26}$$

Because $(\widehat{d}_k, \widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k)$ is the optimal solution of (26), we have

$$(2n)^{-1}\|\widetilde{\mathbf{Y}}_k - \mathbf{X}\widehat{\mathbf{C}}_k\|_F^2 + n^{-1}\sum_{\ell=0}^{k-1}\langle\mathbf{X}\widehat{\mathbf{C}}_k, \mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell\rangle + \lambda_k\|\widehat{\mathbf{C}}_k\|_1$$

$$\leq (2n)^{-1}\|\widetilde{\mathbf{Y}}_k - \mathbf{X}\mathbf{C}_k^*\|_F^2 + n^{-1}\sum_{\ell=0}^{k-1}\langle\mathbf{X}\mathbf{C}_k^*, \mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell\rangle + \lambda_k\|\mathbf{C}_k^*\|_1.$$

Plugging in $\widetilde{\mathbf{Y}}_k - \mathbf{X}\widehat{\mathbf{C}}_k = \widetilde{\mathbf{Y}}_k - \mathbf{X}\mathbf{C}_k^* - \mathbf{X}\widehat{\boldsymbol{\Delta}}_k$, direct calculation yields that

$$(2n)^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2 + n^{-1}\sum_{\ell=0}^{k-1}\langle \mathbf{X}\widehat{\boldsymbol{\Delta}}_k, \mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell\rangle + \lambda_k\|\widehat{\mathbf{C}}_k\|_1$$

$$\leq n^{-1}\langle \mathbf{X}\widehat{\boldsymbol{\Delta}}_k, \mathbf{E}\rangle + n^{-1}\sum_{\ell=k+1}^{r^*}\langle \mathbf{X}\widehat{\boldsymbol{\Delta}}_k, \mathbf{X}\mathbf{C}_\ell^*\rangle + \lambda_k\|\mathbf{C}_k^*\|_1.$$

Utilizing Lemma 1, we can derive that for $k = 2, \ldots, r^*$,

$$(2n)^{-1}(1-\omega_k)\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2 + n^{-1}\sum_{\ell=0}^{k-1}\langle \mathbf{X}\widehat{\boldsymbol{\Delta}}_k, \mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell\rangle + \lambda_k\|\widehat{\mathbf{C}}_k\|_1 \tag{27}$$
$$\leq n^{-1}\langle \mathbf{X}\widehat{\boldsymbol{\Delta}}_k, \mathbf{E}\rangle + \lambda_k\|\mathbf{C}_k^*\|_1.$$

Hereafter, our discussion will be conditioning on the following event

$$\mathcal{A} = \{n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{E}\|_{\max} \leq \lambda_1/2\},$$

which has been shown to hold with a significant probability in the proof of Lemma 2.

We then prove the results for $k > 1$ by mathematical induction. Assume that

$$\|\widehat{\boldsymbol{\Delta}}_{J_\ell^c}\|_1 \leq 3\|\widehat{\boldsymbol{\Delta}}_{J_\ell}\|_1, \quad \|\widehat{\boldsymbol{\Delta}}_{J_\ell}\|_F \leq M_\ell\sqrt{s_\ell}\lambda_\ell,$$
$$\|\widehat{\boldsymbol{\Delta}}_{\tilde{J}_\ell^c, s_\ell}\|_F \leq 3M_\ell\sqrt{s_\ell}\lambda_\ell, \quad \|\widehat{\boldsymbol{\Delta}}_{J_\ell^c, s_\ell}\|_F \leq M_\ell\sqrt{s_\ell}\lambda_\ell$$

hold for $1 \leq \ell \leq k - 1$, where $M_\ell = 12(\rho_l - \rho_l\omega_\ell)^{-1}$, $\widehat{\boldsymbol{\Delta}}_{J_\ell^c, s_\ell}$ is the submatrix of $\widehat{\boldsymbol{\Delta}}_{J_\ell^c}$ as defined in Condition 2.3, and $\widehat{\boldsymbol{\Delta}}_{\tilde{J}_\ell^c, s_\ell}$ is the submatrix of $\widehat{\boldsymbol{\Delta}}_{J_\ell^c}$ consisting of the components excluding those with the s_ℓ largest absolute values.

First of all, by Condition 2.4, we get

$$n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\widehat{\boldsymbol{\Delta}}_\ell\|_{\max} \leq s_\ell^{-1/2}\phi_u\|\widehat{\boldsymbol{\Delta}}_{J_\ell}\|_F \leq \phi_u M_\ell\lambda_\ell,$$

where the last inequality holds because of $\|\widehat{\boldsymbol{\Delta}}_{J_\ell}\|_F \leq M_\ell\sqrt{s_\ell}\lambda_\ell$. Define the kth regularization parameter inductively as

$$\lambda_k = 2(2^{-1}\lambda_1 + \phi_u\sum_{\ell=1}^{k-1}M_\ell\lambda_\ell) = \lambda_1 + 2\phi_u M\sum_{\ell=1}^{k-1}\theta_\ell\lambda_\ell = \lambda_1 + \sum_{\ell=1}^{k-1}\eta_\ell\lambda_\ell, \tag{28}$$

where $M = 12\rho_l^{-1}$, $\theta_\ell = (1-\omega_\ell)^{-1} = d_\ell^*/\delta_\ell^*$, and $\eta_\ell = 24\phi_u\rho_l^{-1}\theta_\ell = c\theta_\ell$ with $c = 24\phi_u\rho_l^{-1}$. Then by the same argument as inequalities (18) and (19) in the proof of Lemma 2, we can get

$$\|\widehat{\boldsymbol{\Delta}}_{J_k^c}\|_1 \leq 3\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_1. \tag{29}$$

Further applying Condition 2.3 gives

$$\rho_l(\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F^2 \vee \|\widehat{\boldsymbol{\Delta}}_{J_k^c, s_k}\|_F^2) \leq n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2.$$

Then by inequality (27) and the same argument as inequality (21), we have

$$(2n)^{-1}(1 - \omega_k)\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2 \leq \frac{3}{2}\lambda_k\|\widehat{\boldsymbol{\Delta}}_k\|_1 \leq 6\lambda_k\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_1 \leq 6\lambda_k\sqrt{s_k}\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F.$$

Combining these two inequalities yields that

$$\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F \leq 12(\rho_l - \rho_l\omega_k)^{-1}\sqrt{s_k}\lambda_k = M_k\sqrt{s_k}\lambda_k. \tag{30}$$

Then by the same argument as inequalities (23) and (24), we can get

$$\|\widehat{\boldsymbol{\Delta}}_{\tilde{J}_{k,s_k}^c}\|_F \leq s_k^{-1/2}\|\widehat{\boldsymbol{\Delta}}_{J_k^c}\|_1 \leq 3s_k^{-1/2}\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_1 \leq 3\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F,$$

$$\|\widehat{\boldsymbol{\Delta}}_{J_{k,s_k}^c}\|_F \leq \{12\lambda_k(\rho_l - \rho_l\omega_k)^{-1}\sqrt{s_k}\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F\}^{1/2}.$$

It yields that

$$\begin{aligned}
\|\widehat{\boldsymbol{\Delta}}_{\tilde{J}_{k,s_k}^c}\|_F &\leq 36(\rho_l - \rho_l\omega_k)^{-1}\sqrt{s_k}\lambda_k = 3M_k\sqrt{s_k}\lambda_k, \\
\|\widehat{\boldsymbol{\Delta}}_{J_{k,s_k}^c}\|_F &\leq 12(\rho_l - \rho_l\omega_k)^{-1}\sqrt{s_k}\lambda_k = M_k\sqrt{s_k}\lambda_k.
\end{aligned} \tag{31}$$

Therefore, we can derive that inequalities (29)–(31) hold for $2 \leq k \leq r^*$ by mathematical induction. By the same argument as that in the proof of Lemma 2, inequalities (29)–(31) give

$$\|\widehat{\boldsymbol{\Delta}}_k\|_F \leq \|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F + \|\widehat{\boldsymbol{\Delta}}_{J_{k,s_k}^c}\|_F + \|\widehat{\boldsymbol{\Delta}}_{\tilde{J}_{k,s_k}^c}\|_F \leq 5M_k\sqrt{s_k}\lambda_k = O(\theta_k\sqrt{s_k}\lambda_k),$$

$$\|\widehat{\boldsymbol{\Delta}}_k\|_1 \leq 4\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_1 \leq 4\sqrt{s_k}\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F \leq 4M_ks_k\lambda_k = O(\theta_k s_k\lambda_k).$$

Finally, we derive an explicit form of the regularization parameter λ_k defined in (28). By definition, we have

$$\lambda_k = \lambda_1 + \sum_{\ell=1}^{k-1}\eta_\ell\lambda_\ell, \quad \lambda_{k-1} = \lambda_1 + \sum_{\ell=1}^{k-2}\eta_\ell\lambda_\ell.$$

It yields that

$$\lambda_k - \lambda_{k-1} = \eta_{k-1}\lambda_{k-1},$$

which gives

$$\lambda_k = (1 + \eta_{k-1})\lambda_{k-1}.$$

Thus, we can derive $\lambda_k = \Pi_{\ell=1}^{k-1}(1 + \eta_\ell)\lambda_1$ by mathematical induction, which concludes the proof of Theorem 1.

B. Proof for the parallel pursuit

Recall that the parallel pursuit is as follows

$$\begin{aligned}
(\widehat{d}_k, \widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k) = \arg\min_{(d,\mathbf{u},\mathbf{v})} & \ (2n)^{-1}\|\mathbf{Y}_k - d\mathbf{X}\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_F^2 + \lambda_k\|d\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_1 \\
\text{s.t. } & d > 0, n^{-1}\mathbf{u}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{u} = \mathbf{v}^{\mathrm{T}}\mathbf{v} = 1.
\end{aligned} \tag{32}$$

where $\mathbf{Y}_k = \mathbf{Y} - \mathbf{X}\sum_{\ell \neq k}\widetilde{\mathbf{C}}_\ell$ with the initial ℓth layer estimator $\widetilde{\mathbf{C}}_\ell$ that by definition takes the s largest components of $\widetilde{\mathbf{C}}_k^0$ in terms of absolute values while sets the others to be zero. Here $\widetilde{\mathbf{C}}_\ell^0$ is $\ell$the unit rank matrix of the initial Lasso estimator $\widetilde{\mathbf{C}}$, which is generated from

$$\widetilde{\mathbf{C}} = \arg\min_{\mathbf{C}} \ (2n)^{-1}\|\mathbf{Y} - \mathbf{X}\mathbf{C}\|_F^2 + \lambda_0\|\mathbf{C}\|_1. \tag{33}$$

We need two lemmas before showing the main results of the parallel pursuit.

B.1 Lemma 3 and its proof

Lemma 3 *When Conditions 2.1 and 2.3 hold with the sparsity level $s \geq s_0$, $\lambda_0 = 2\sigma_{\max}\sqrt{2\alpha \log(pq)/n}$ for some constant $\alpha > 1$, we have with probability at least $1 - (pq)^{1-\alpha}$, the initial Lasso estimator $\widetilde{\mathbf{C}}$ satisfies*

$$\|\widetilde{\boldsymbol{\Delta}}\|_F = O(\sqrt{s_0 \log(pq)/n}), \ \ n^{-1/2}\|\mathbf{X}\widetilde{\boldsymbol{\Delta}}\|_F = O(\sqrt{s_0 \log(pq)/n}),$$

where $s_0 = \|\mathbf{C}^*\|_0$ and $\widetilde{\boldsymbol{\Delta}} = \widetilde{\mathbf{C}} - \mathbf{C}^*$.

Proof [Proof of Lemma 3] By the optimality of $\widetilde{\mathbf{C}}$, we have

$$(2n)^{-1}\|\mathbf{X}\widetilde{\boldsymbol{\Delta}}\|_F^2 + \lambda_0\|\widetilde{\mathbf{C}}\|_1 \leq \langle \widetilde{\boldsymbol{\Delta}}, \mathbf{X}^{\mathrm{T}}\mathbf{E}\rangle + \lambda_0\|\mathbf{C}^*\|_1.$$

Applying the same argument as inequalities (18) and (19) in the proof of Lemma 2, conditioning on the event $\mathcal{A} = \{\|\mathbf{X}^{\mathrm{T}}\mathbf{E}\|_{\max} \leq \lambda_0/2\}$, we can get

$$\|\widetilde{\boldsymbol{\Delta}}_{J^c}\|_1 \leq 3\|\widetilde{\boldsymbol{\Delta}}_J\|_1.$$

Combining these two inequalities gives that

$$n^{-1}\|\mathbf{X}\widetilde{\boldsymbol{\Delta}}\|_F^2 \leq 3\lambda_0\|\widetilde{\boldsymbol{\Delta}}\|_1 \leq 12\lambda_0\|\widetilde{\boldsymbol{\Delta}}_J\|_1 \leq 12\sqrt{s_0}\lambda_0\|\widetilde{\boldsymbol{\Delta}}_J\|_F. \tag{34}$$

On the other hand, by Condition 2.3, we have

$$\rho_l(\|\widetilde{\boldsymbol{\Delta}}_J\|_F^2 \vee \|\widetilde{\boldsymbol{\Delta}}_{J_{s_0}^c}\|_F^2) \leq n^{-1}\|\mathbf{X}\widetilde{\boldsymbol{\Delta}}\|_F^2. \tag{35}$$

Therefore, together with (34), it yields

$$\|\widetilde{\boldsymbol{\Delta}}_J\|_F \leq 12\rho_l^{-1}\sqrt{s_0}\lambda_0, \ \ \|\widetilde{\boldsymbol{\Delta}}_{J_{s_0}^c}\|_F \leq 12\rho_l^{-1}\sqrt{s_0}\lambda_0.$$

By the same argument as that in the proof of Lemma 2, when $\lambda_0 = 2\sigma_{\max}\sqrt{2\alpha \log(pq)/n}$, the following results hold with probability at least $1 - (pq)^{1-\alpha}$,

$$\|\widetilde{\boldsymbol{\Delta}}\|_F = O(\sqrt{s_0}\lambda_0), \ \ n^{-1/2}\|\mathbf{X}\widetilde{\boldsymbol{\Delta}}\|_F = O(\sqrt{s_0}\lambda_0).$$

It concludes the proof of Lemma 3. ∎

B.2 Lemma 4 and its proof

Lemma 4 *When $\lambda_0 = 2\sigma_{\max}\sqrt{2\alpha \log(pq)/n}$ for some constant $\alpha > 1$, Conditions 2.1–2.3 hold with the sparsity level $s \geq s_0$, then there exists some positive constant M such that with probability at least $1 - (pq)^{1-\alpha}$, the following inequality holds uniformly over $1 \leq k \leq r^*$,*

$$\|\widetilde{\mathbf{C}}_k^0 - \mathbf{C}_k^*\|_F \leq M\psi_k\sqrt{s_0 \log(pq)/n},$$

where $\psi_k = d_1^* d_c^* / (d_k^* \min[\delta_{k-1}^*, \delta_k^*])$ with d_c^* the largest singular value of $\mathbf{C}^*$.

Proof [Proof of Lemma 4]

Note that $\widetilde{\mathbf{v}}_k$ and $\mathbf{v}_k^*$ are also the unit length right singular vectors of $\mathbf{X}\widetilde{\mathbf{C}}$ and $\mathbf{X}\mathbf{C}^*$, respectively, $\widetilde{\mathbf{v}}_k^T \mathbf{v}_k^* \geq 0$, and we have $\widetilde{d}_k \widetilde{\mathbf{u}}_k = \widetilde{\mathbf{C}}\widetilde{\mathbf{v}}_k$ and $d_k^* \mathbf{u}_k^* = \mathbf{C}^* \mathbf{v}_k^*$. It follows that

$$
\begin{aligned}
\widetilde{\mathbf{C}}_k^0 - \mathbf{C}_k^* &= \widetilde{d}_k \widetilde{\mathbf{u}}_k \widetilde{\mathbf{v}}_k^T - d_k^* \mathbf{u}_k^* \mathbf{v}_k^{*T} = d_k^* \mathbf{u}_k^* (\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*)^T + (\widetilde{d}_k \widetilde{\mathbf{u}}_k - d_k^* \mathbf{u}_k^*)\widetilde{\mathbf{v}}_k^T \\
&= d_k^* \mathbf{u}_k^* (\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*)^T + \left[\mathbf{C}^*(\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*) + (\widetilde{\mathbf{C}} - \mathbf{C}^*)\widetilde{\mathbf{v}}_k \right] \widetilde{\mathbf{v}}_k^T.
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
\|\widetilde{\mathbf{C}}_k^0 - \mathbf{C}_k^*\|_F &\leq d_k^* \|\mathbf{u}_k^*\|_2 \|\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 + \|\mathbf{C}^*(\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*)\|_2 + \|(\widetilde{\mathbf{C}} - \mathbf{C}^*)\widetilde{\mathbf{v}}_k\|_2 \\
&\leq d_k^* \|\mathbf{u}_k^*\|_2 \|\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 + d_c^* \|\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 + \|\widetilde{\mathbf{C}} - \mathbf{C}^*\|_F \qquad (36) \\
&\leq 2d_c^* \|\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 + \|\widetilde{\mathbf{C}} - \mathbf{C}^*\|_F,
\end{aligned}
$$

where d_c^* is the largest singular value of $\mathbf{C}^*$ and the last inequality makes use of the fact that $d_k^* \|\mathbf{u}_k^*\|_2 \leq d_c^*$ for any k, $1 \leq k \leq r^*$, due to the following inequality

$$
d_k^* \|\mathbf{u}_k^*\|_2 = \frac{\mathbf{u}_k^* \mathbf{C}^* \mathbf{v}_k *}{\|\mathbf{u}_k^*\|_2} \leq \max_{\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1} \mathbf{u}^T \mathbf{C}^* \mathbf{v} = d_c^*.
$$

Moreover, applying Yu et al. (2014, Theorem 3), we get

$$
\|\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 \leq \frac{2^{3/2}(2d_1^* + n^{-1/2}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}\|_F)n^{-1/2}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}\|_F}{\min(d_{k-1}^{*2} - d_k^{*2}, d_k^{*2} - d_{k+1}^{*2})}
$$

for any k, $1 \leq k \leq r^*$ with $d_0^* = +\infty$. By Condition 2.2, we have

$$
\begin{aligned}
d_{k-1}^{*2} - d_k^{*2} &= (d_{k-1}^* + d_k^*)(d_{k-1}^* - d_k^*) = (d_{k-1}^* + d_k^*)\delta_{k-1}^* > d_k^* \delta_{k-1}^*, \\
d_k^{*2} - d_{k+1}^{*2} &= (d_k^* + d_{k+1}^*)(d_k^* - d_{k+1}^*) = (d_k^* + d_{k+1}^*)\delta_k^* > d_k^* \delta_k^*,
\end{aligned}
$$

which yields

$$
\min(d_{k-1}^{*2} - d_k^{*2}, d_k^{*2} - d_{k+1}^{*2}) > d_k^* \min[\delta_{k-1}^*, \delta_k^*].
$$

Then together with the results of Lemma 3, it follows that

$$
\|\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 = O\left\{ \frac{d_1^* n^{-1/2}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}\|_F}{d_k^* \min[\delta_{k-1}^*, \delta_k^*]} \right\} = O(\widetilde{\psi}_k \sqrt{s_0 \log(pq)/n}), \qquad (37)
$$

uniformly over $1 \leq k \leq r^*$, where $\widetilde{\psi}_k = d_1^*/(d_k^* \min[\delta_{k-1}^*, \delta_k^*])$. Combining inequalities (36) and (37) and Lemma 3 entails

$$
\|\widetilde{\mathbf{C}}_k^0 - \mathbf{C}_k^*\|_F \leq 2d_c^* \|\widetilde{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 + \|\widetilde{\mathbf{C}} - \mathbf{C}^*\|_F \leq M\psi_k \sqrt{s_0 \log(pq)/n}
$$

for some positive constant M, uniformly over $1 \leq k \leq r^*$, where $\psi_k = d_1^* d_c^*/(d_k^* \min[\delta_{k-1}^*, \delta_k^*])$. It concludes the proof of Lemma 4. ∎

33

B.3 Proof of Theorem 2

Proof Since $(\widehat{d}_k, \widehat{\mathbf{u}}_k, \widehat{\mathbf{v}}_k)$ is the optimal solution of (32), we have

$$(2n)^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F + \lambda_k\|\widehat{\mathbf{C}}_k\|_1 \leq n^{-1}\langle\widehat{\boldsymbol{\Delta}}_k, \mathbf{X}^{\mathrm{T}}(\mathbf{Y}_k - \mathbf{X}\mathbf{C}_k^*)\rangle + \lambda_k\|\mathbf{C}_k^*\|_1 \tag{38}$$

with $\mathbf{Y}_k = \mathbf{Y} - \mathbf{X}\widetilde{\mathcal{C}} + \mathbf{X}\widetilde{\mathbf{C}}_k$. Our proof will be conditioning on the event $\mathcal{A} = \{n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{E}\|_{\max} \leq \lambda_0/2\}$, which holds with probability at least $1 - (pq)^{1-\alpha}$ as demonstrated in Lemma 3.

The key part of this proof is to derive the upper bound on $n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y}_k - \mathbf{X}\mathbf{C}_k^*)\|_{\max}$. On the one hand, by the KKT condition of optimization (33), we have

$$n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\widetilde{\mathcal{C}})\|_{\max} \leq \lambda_0 = 2\sigma_{\max}\sqrt{2\alpha\log(pq)/n}. \tag{39}$$

On the other hand, Lemma 4 gives

$$\|\widetilde{\mathbf{C}}_k^0 - \mathbf{C}_k^*\|_F \leq M\psi_k\sqrt{s_0\log(pq)/n}.$$

Denote by S_k the index set consisting of the indices of the s largest components of $\widetilde{\mathbf{C}}_k^0$ in terms of absolute values. By definition, our initial kth layer estimator $\widetilde{\mathbf{C}}_k$ takes the components of $\widetilde{\mathbf{C}}_k^0$ on S_k while sets the other components to be zero.

First of all, we claim that

$$\|\mathbf{C}_{k,J_k\cap S_k^C}^*\|_F \leq 2M\psi_k\sqrt{s_0\log(pq)/n}. \tag{40}$$

If this is not true, then $\|\mathbf{C}_{k,J_k\cap S_k^C}^*\|_F > 2M\psi_k\sqrt{s_0\log(pq)/n}$. We can deduce that

$$\begin{aligned}
\|\widetilde{\mathbf{C}}_{k,J_k\cap S_k^C}^0\|_F &= \|\widetilde{\mathbf{C}}_{k,J_k\cap S_k^C}^0 - \mathbf{C}_{k,J_k\cap S_k^C}^* + \mathbf{C}_{k,J_k\cap S_k^C}^*\|_F \\
&\geq \|\mathbf{C}_{k,J_k\cap S_k^C}^*\|_F - \|\widetilde{\mathbf{C}}_{k,J_k\cap S_k^C}^0 - \mathbf{C}_{k,J_k\cap S_k^C}^*\|_F \\
&\geq \|\mathbf{C}_{k,J_k\cap S_k^C}^*\|_F - \|\widetilde{\mathbf{C}}_k^0 - \mathbf{C}_k^*\|_F > M\psi_k\sqrt{s_0\log(pq)/n}.
\end{aligned}$$

On the other hand, since J_k is the support of $\mathbf{C}_k^*$, we have

$$\begin{aligned}
\|\widetilde{\mathbf{C}}_{k,S_k\cap J_k^C}^0\|_F &= \|\widetilde{\mathbf{C}}_{k,J_k\cap S_k^C}^0 - \mathbf{C}_{k,S_k\cap J_k^C}^* + \mathbf{C}_{k,S_k\cap J_k^C}^*\|_F \\
&\leq \|\mathbf{C}_{k,S_k\cap J_k^C}^*\|_F + \|\widetilde{\mathbf{C}}_{k,S_k\cap J_k^C}^0 - \mathbf{C}_{k,S_k\cap J_k^C}^*\|_F \\
&\leq \|\mathbf{C}_{k,S_k\cap J_k^C}^*\|_F + \|\widetilde{\mathbf{C}}_k^0 - \mathbf{C}_k^*\|_F \leq M\psi_k\sqrt{s_0\log(pq)/n}.
\end{aligned}$$

In view of these two inequalities, we get

$$\|\widetilde{\mathbf{C}}_{k,J_k\cap S_k^C}^0\|_F > \|\widetilde{\mathbf{C}}_{k,S_k\cap J_k^C}^0\|_F.$$

This is a contraction since by the definition of S_k, the set $J_k \cap S_k^C$ contains $s_k^* - t$ elements with $t = |J_k \cap S_k|$, each of which in $\widetilde{\mathbf{C}}_k^0$ should be no larger than any of the $s - t$ components indexed by $S_k \cap J_k^C$, and $s - t$ is no less than $s_k^* - t$. Therefore, we know that inequality (40) should be true. It follows that

$$\begin{aligned}
\|\widetilde{\mathbf{C}}_k - \mathbf{C}_k^*\|_F &\leq \|\widetilde{\mathbf{C}}_{k,S_k} - \mathbf{C}_{k,S_k}^*\|_F + \|\mathbf{C}_{k,J_k\cap S_k^C}^*\|_F \\
&\leq \|\widetilde{\mathbf{C}}_k^0 - \mathbf{C}_k^*\|_F + \|\mathbf{C}_{k,J_k\cap S_k^C}^*\|_F \leq 3M\psi_k\sqrt{s_0\log(pq)/n}.
\end{aligned} \tag{41}$$

Moreover, since $\|\widetilde{\mathbf{C}}_k\|_0 = s$ and $\|\mathbf{C}_k^*\|_0 = s_k^* \leq s$, we have

$$\|\widetilde{\boldsymbol{\Delta}}_k\|_0 = \|\widetilde{\mathbf{C}}_k - \mathbf{C}_k^*\|_0 \leq \|\widetilde{\mathbf{C}}_k\|_0 + \|\mathbf{C}_k^*\|_0 = s + s_k^* \leq 2s.$$

Denote by $\widetilde{J}_k$ the index set consisting of the indices of the s largest components of $\widetilde{\boldsymbol{\Delta}}_k$ in terms of absolute values. It is clear that $\|\widetilde{\boldsymbol{\Delta}}_{\widetilde{J}_k^c}\|_1 \leq \|\widetilde{\boldsymbol{\Delta}}_{\widetilde{J}_k}\|_1$. Thus, by Condition 2.4, we get

$$n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\widetilde{\boldsymbol{\Delta}}_k\|_{\max} \leq \phi_u \|\widetilde{\boldsymbol{\Delta}}_{\widetilde{J}_k}\|_F/\sqrt{s} \leq \phi_u \|\widetilde{\boldsymbol{\Delta}}_k\|_F/\sqrt{s}. \tag{42}$$

As $s \geq s_0$, combining inequalities (41) and (42) gives

$$n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\widetilde{\boldsymbol{\Delta}}_k\|_{\max} \leq 3M\phi_u\psi_k\sqrt{\log(pq)/n}.$$

Together with (39), we can derive that

$$n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y}_k - \mathbf{X}\mathbf{C}_k^*)\|_{\max} \leq n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\widetilde{\mathbf{C}})\|_{\max} + n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\widetilde{\boldsymbol{\Delta}}_k\|_{\max} \leq C\psi_k\sqrt{\log(pq)/n}$$

for some positive constant C independent of k.

Therefore, when $\lambda_k = 2C\psi_k\sqrt{\log(pq)/n}$, by the same argument as inequality (29), we get

$$\|\widehat{\boldsymbol{\Delta}}_{J_k^c}\|_1 \leq 3\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_1. \tag{43}$$

Similarly, it follows from inequality (38) that

$$n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2 \leq 3\lambda_k\|\widehat{\boldsymbol{\Delta}}_k\|_1 \leq 12\lambda_k\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_1 \leq 12\lambda_k\sqrt{s_k}\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F.$$

Applying Condition 2.3, we have

$$\rho_l(\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F^2 \vee \|\widehat{\boldsymbol{\Delta}}_{J_k^c,s_k}\|_F^2) \leq n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}_k\|_F^2 \leq 12\lambda_k\sqrt{s_k}\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F. \tag{44}$$

Finally, by the same argument as that in the proof of Lemma 2, we can get

$$\begin{aligned}
\|\widehat{\boldsymbol{\Delta}}_k\|_F &\leq \|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F + \|\widehat{\boldsymbol{\Delta}}_{J_k^c,s_k}\|_F + \|\widehat{\boldsymbol{\Delta}}_{\widetilde{J}_k^c,s_k}\|_F = O(\lambda_k\sqrt{s_k}), \\
\|\widehat{\boldsymbol{\Delta}}_k\|_1 &= \|\widehat{\boldsymbol{\Delta}}_{J_k}\|_1 + \|\widehat{\boldsymbol{\Delta}}_{J_k^c}\|_1 \leq 4\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_1 \leq 4\sqrt{s_k}\|\widehat{\boldsymbol{\Delta}}_{J_k}\|_F = O(\lambda_k s_k).
\end{aligned} \tag{45}$$

Since the above argument applies to any fixed k, $1 \leq k \leq r^*$, and the corresponding constants are independent of k, we know that the estimation error bounds in (45) hold uniformly over k. It concludes the proof of Theorem 2. ∎

C. Proof of Theorem 3

Note that $\widehat{\mathbf{C}}_k$ is the global minimizer of the corresponding CURE problem and $\widehat{\mathbf{C}}_k^L$ is a local minimizer that satisfies the conditions in Theorem 3. Since the following argument applies to any fixed k, $1 \leq k \leq r^*$, we omit the subscript k and write them as $\widehat{\mathbf{C}}$ and $\widehat{\mathbf{C}}^L$ for simplicity. We will show that $\|\boldsymbol{\Delta}\|_F = O(s_k^{1/2}\lambda_k)$ and $\|\boldsymbol{\Delta}\|_1 = O(s_k\lambda_k)$ with $\boldsymbol{\Delta} = \widehat{\mathbf{C}}^L - \widehat{\mathbf{C}}$ such that $\widehat{\mathbf{C}}^L$ enjoys the same asymptotic properties as $\widehat{\mathbf{C}}$ in terms of estimation error bounds.

Our proof will be conditioning on the event $\mathcal{A}$ (defined in Lemma 2 for the sequential pursuit and defined in Lemma 3 for the parallel pursuit), which holds with probability at least $1 - (pq)^{1-\alpha}$. First of all, by the KKT condition, the global minimizer $\widehat{\mathbf{C}}$ satisfies

$$n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y}_k - \mathbf{X}\widehat{\mathbf{C}})\|_{\max} \leq \lambda_k.$$

Together with $n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y}_k - \mathbf{X}\widehat{\mathbf{C}}^L)\|_{\max} = O(\lambda_k)$ imposed in Theorem 3, we get

$$n^{-1}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\Delta}\|_{\max} \leq n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y}_k - \mathbf{X}\widehat{\mathbf{C}})\|_{\max} + n^{-1}\|\mathbf{X}^{\mathrm{T}}(\mathbf{Y}_k - \mathbf{X}\widehat{\mathbf{C}}^L)\|_{\max} = O(\lambda_k).$$

Denote by $A_j = \mathrm{supp}(\boldsymbol{\Delta}_j)$ with $\boldsymbol{\Delta}_j$ being the jth column of $\boldsymbol{\Delta}$. It follows that

$$n^{-1}\|\mathbf{X}_{A_j}^{\mathrm{T}}\mathbf{X}_{A_j}\boldsymbol{\Delta}_{A_j}\|_2 \leq n^{-1}\sqrt{|A_j|}\|\mathbf{X}_{A_j}^{\mathrm{T}}\mathbf{X}_{A_j}\boldsymbol{\Delta}_{A_j}\|_{\max} \leq n^{-1}\sqrt{|A_j|}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}_{A_j}\boldsymbol{\Delta}_{A_j}\|_{\max}$$
$$= n^{-1}\sqrt{|A_j|}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\Delta}_j\|_{\max} \leq n^{-1}\sqrt{|A_j|}\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\Delta}\|_{\max} = O(|A_j|^{1/2}\lambda_k),$$

where $\mathbf{X}_{A_j}$ ($\boldsymbol{\Delta}_{A_j}$) is the submatrix (subvector) of $\mathbf{X}$ ($\boldsymbol{\Delta}_j$) consisting of columns (components) in A_j.

Moreover, by the assumptions $\|\widehat{\mathbf{C}}\|_0 = O(s_k)$ and $\|\widehat{\mathbf{C}}^L\|_0 = O(s_k)$, there exists some positive constant C such that

$$\sum_{j=1}^{q} |A_j| \leq \|\widehat{\mathbf{C}}\|_0 + \|\widehat{\mathbf{C}}^L\|_0 \leq Cs_k.$$

Therefore, it follows from the assumption $\min_{\|\boldsymbol{\gamma}\|_2=1, \|\boldsymbol{\gamma}\|_0 \leq Cs_k} n^{-1/2}\|\mathbf{X}\boldsymbol{\gamma}\|_2 \geq \kappa_0$ in Theorem 3 that the smallest singular value of $n^{-1/2}\mathbf{X}_{A_j}$ is bounded from below by κ_0. It yields

$$\kappa_0^2\|\boldsymbol{\Delta}_{A_j}\|_2 \leq n^{-1}\|\mathbf{X}_{A_j}^{\mathrm{T}}\mathbf{X}_{A_j}\boldsymbol{\Delta}_{A_j}\|_2 = O(|A_j|^{1/2}\lambda_k).$$

Since $\|\boldsymbol{\Delta}\|_F^2 = \sum_{j=1}^{q}\|\boldsymbol{\Delta}_j\|_2^2 = \sum_{j=1}^{q}\|\boldsymbol{\Delta}_{A_j}\|_2^2$, we finally get

$$\|\boldsymbol{\Delta}\|_F^2 = O(\lambda_k^2 \sum_{j=1}^{q} |A_j|) = O(s_k\lambda_k^2).$$

It follows immediately that $\|\boldsymbol{\Delta}\|_F = O(s_k^{1/2}\lambda_k)$ and $\|\boldsymbol{\Delta}\|_1 = O(s_k\lambda_k)$, which concludes the proof.

D. Proof of Stagewise CURE

D.1 Derivations of contended stagewise learning

Before presenting the derivations, we first recall some notations for the sake of clarity. Denote $\mathbf{X} = [\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_p] = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times p}$ and $\mathbf{Y} = [\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_q] = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times q}$ then let $\widetilde{\mathbf{x}}_j$ and $\widetilde{\mathbf{y}}_k$ be the jth and kth columns of $\mathbf{X}$ and $\mathbf{Y}$, respectively. In addition, let $\mathbf{E}^t$ be equal to $\mathbf{Y} - d^t\mathbf{X}\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}$ and $\widetilde{\mathbf{e}}_k$ denotes the kth column of $\mathbf{E}^t$.

(I) Initialization. Recall that the optimization in the initialization is as follow

$$(\widehat{j}, \widehat{k}, \widehat{s}) = \underset{(j,k); s=\pm\epsilon}{\arg\min} \; L(s\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}}).$$

By the expansion of the loss function, we have

$$L(s\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}}) = (2n)^{-1}\|\mathbf{Y} - s\mathbf{X}\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}}\|_F^2 + \frac{\mu}{2}\|s\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}}\|_F^2$$

$$= (2n)^{-1}\|\mathbf{Y}\|_F^2 + (2n)^{-1}\epsilon^2\mathbf{1}_j^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{1}_j - n^{-1}s\mathbf{1}_j^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{Y}\mathbf{1}_k + \frac{\mu}{2}\epsilon^2.$$

Thus the original problem is equivalent to

$$(\widehat{j}, \widehat{k}) = \arg\min_{j,k} (2n)^{-1}\epsilon\|\widetilde{\mathbf{x}}_j\|_2^2 - n^{-1}|\widetilde{\mathbf{x}}_j^{\mathrm{T}}\widetilde{\mathbf{y}}_k|, \ \ \widehat{s} = \mathrm{sgn}(\widetilde{\mathbf{x}}_j^{\mathrm{T}}\widetilde{\mathbf{y}}_k)\epsilon.$$

(II) Backward update. At the $(t+1)$th step, the two updating options are as follows,

$$\widehat{j} = \arg\min_{j\in\mathcal{A}^t} L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) \text{ s.t. } (d\mathbf{u}) = (d\mathbf{u})^t - \mathrm{sgn}(u_j^t)\epsilon\mathbf{1}_j, \mathbf{v} = \mathbf{v}^t,$$

$$\widehat{k} = \arg\min_{k\in\mathcal{B}^t} L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) \text{ s.t. } (d\mathbf{v}) = (d\mathbf{v})^t - \mathrm{sgn}(v_k^t)\epsilon\mathbf{1}_k, \mathbf{u} = \mathbf{u}^t.$$

Note that, either updating $d\mathbf{u}$ or $d\mathbf{v}$ will decrease the penalty term by a fixed amount $\lambda^t\epsilon$. For example, assuming that we update $d\mathbf{u}$, then

$$\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \rho(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) = \|(d\mathbf{u})^t - \mathrm{sgn}(u_{\widehat{j}}^t)\epsilon\mathbf{1}_{\widehat{j}}\|_1\|\mathbf{v}^t\|_1 - d^t\|\mathbf{u}^t\|_1\|\mathbf{v}^t\|_1$$

$$= \|(d\mathbf{u})^t - \mathrm{sgn}(u_{\widehat{j}}^t)\epsilon\mathbf{1}_{\widehat{j}}\|_1 - \|(d\mathbf{u})^t\|_1 = -\epsilon.$$

It follows that we only need to compare the value of the loss function to decide whether to update $d\mathbf{u}$ or $d\mathbf{v}$.

Focusing on $d\mathbf{u}$, the loss function can be written as

$$L\left(\left[(d\mathbf{u})^t - \mathrm{sgn}(u_j^t)\epsilon\mathbf{1}_j\right]\mathbf{v}^{t\mathrm{T}}\right) = (2n)^{-1}\|\mathbf{Y} - \mathbf{X}\left[(d\mathbf{u})^t - \mathrm{sgn}(u_j^t)\epsilon\mathbf{1}_j\right]\mathbf{v}^{t\mathrm{T}}\|_F^2 +$$

$$\frac{\mu}{2}\|\mathbf{v}^t\|_2^2\|(d\mathbf{u})^t - \mathrm{sgn}(u_j^t)\epsilon\mathbf{1}_j\|_2^2$$

$$= (2n)^{-1}\epsilon^2\|\widetilde{\mathbf{x}}_j\|_2^2\|\mathbf{v}^t\|_2^2 + n^{-1}\mathrm{sgn}(u_j^t)\epsilon\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t \qquad (46)$$

$$- \mu\epsilon d^t|u_j^t|\|\mathbf{v}^t\|_2^2 + \frac{\mu}{2}\epsilon^2\|\mathbf{v}^t\|_2^2 + L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}),$$

where $\mathbf{E}^t = \mathbf{Y} - d^t\mathbf{X}\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}$, and $\widetilde{\mathbf{x}}_j$ is the jth column of $\mathbf{X}$. For $d\mathbf{v}$, with similar argument, we have

$$L\left(\mathbf{u}^t\left[(d\mathbf{v})^t - \mathrm{sgn}(v_k^t)\epsilon\mathbf{1}_k\right]^{\mathrm{T}}\right) = n^{-1}\mathrm{sgn}(v_k^t)\epsilon\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t - \mu\epsilon d^t|v_k^t|\|\mathbf{u}^t\|_2^2 + \frac{\mu}{2}\epsilon^2\|\mathbf{u}^t\|_2^2$$

$$+ (2n)^{-1}\epsilon^2\|\mathbf{X}\mathbf{u}^t\|_2^2 + L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}), \qquad (47)$$

where $\widetilde{\mathbf{e}}_k^t$ is the kth column of $\mathbf{E}^t$. Therefore, the two proposals in the backward step are equivalent to

$$\widehat{j} = \arg\min_{j\in\mathcal{A}^t} (2n)^{-1}\epsilon\|\widetilde{\mathbf{x}}_j\|_2^2\|\mathbf{v}^t\|_2^2 + n^{-1}\mathrm{sgn}(u_j^t)\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t - \mu d^t|u_j^t|\|\mathbf{v}^t\|_2^2,$$

$$\widehat{k} = \arg\min_{k\in\mathcal{B}^t} n^{-1}\mathrm{sgn}(v_k^t)\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t - \mu d^t|v_k^t|\|\mathbf{u}^t\|_2^2. \qquad (48)$$

(III) Forward update. When the backward update can no longer proceed, a forward update is carried out. Let's focus on $d\mathbf{u}$; the problem is

$$\arg\min_{j,s=\pm\epsilon} L\left(\left[(d\mathbf{u})^t + s\mathbf{1}_j\right]\mathbf{v}^{t\mathrm{T}}\right) = (2n)^{-1}\|\mathbf{Y} - \mathbf{X}\left[(d\mathbf{u})^t + s\mathbf{1}_j\right]\mathbf{v}^{t\mathrm{T}}\|_F^2 + \frac{\mu}{2}\|(d\mathbf{u})^t + s\mathbf{1}_j\|_2^2\|\mathbf{v}^t\|_2^2.$$

We have that

$$L\left(\left[(d\mathbf{u})^t + s\mathbf{1}_j\right]\mathbf{v}^{t\mathrm{T}}\right) = (2n)^{-1}\epsilon^2\|\widetilde{\mathbf{x}}_j\|_2^2\|\mathbf{v}^t\|_2^2 - n^{-1}s\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t$$
$$+ \frac{\mu}{2}\epsilon^2\|\mathbf{v}^t\|_2^2 + \mu s d^t u_j^t\|\mathbf{v}^t\|_2^2 + L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}).$$

Similarly, for $d\mathbf{v}$, we have

$$L\left(\mathbf{u}^t\left[(d\mathbf{v})^t + h\mathbf{1}_k\right]^{\mathrm{T}}\right) = \frac{\epsilon^2}{2n}\|\mathbf{X}\mathbf{u}^t\|_2^2 - \frac{h}{n}\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t + \frac{\mu\epsilon^2}{2}\|\mathbf{u}^t\|_2^2 + \mu h d^t v_k^t\|\mathbf{u}^t\|_2^2 + L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}).$$

Therefore, minimizing the loss function is equivalent to

$$\begin{aligned}
&\mathop{\arg\min}_{j,s=\pm\epsilon}\ (2n)^{-1}\epsilon^2\|\widetilde{\mathbf{x}}_j\|_2^2\|\mathbf{v}^t\|_2^2 - s(n^{-1}\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t - \mu d^t u_j^t\|\mathbf{v}^t\|_2^2),\\
&\mathop{\arg\min}_{k,h=\pm\epsilon}\ h(\mu d^t v_k^t\|\mathbf{u}^t\|_2^2 - n^{-1}\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t).
\end{aligned} \tag{49}$$

Then it follows that

$$\begin{aligned}
\widehat{j} &= \mathop{\arg\min}_{j}\ (2n)^{-1}\epsilon\|\widetilde{\mathbf{x}}_j\|_2^2\|\mathbf{v}^t\|_2^2 - |n^{-1}\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t - \mu d^t u_j^t\|\mathbf{v}^t\|_2^2|,\\
\widehat{k} &= \mathop{\arg\min}_{k}\ -|n^{-1}\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t - \mu d^t v_k^t\|\mathbf{u}^t\|_2^2|,
\end{aligned}$$

and

$$\begin{aligned}
\widehat{s} &= \mathrm{sgn}(n^{-1}\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t - \mu d^t u_j^t\|\mathbf{v}^t\|_2^2)\epsilon \text{ for updating } d\mathbf{u};\\
\widehat{h} &= \mathrm{sgn}(n^{-1}\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t - \mu d^t v_k^t\|\mathbf{u}^t\|_2^2)\epsilon \text{ for updating } d\mathbf{v}.
\end{aligned}$$

(IV) Computational complexity. Assume that the tth step is completed, and the cardinalities of active sets $\mathcal{A}^t$ and $\mathcal{B}^t$ are a^t and b^t, respectively. In the $(t+1)$th step, we need to update the current residual matrix $\mathbf{E}^t = \mathbf{Y} - d^t\mathbf{X}\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}$. The complexity for updating $\mathbf{E}^t$ is $O(a^t n + b^t n)$ because we only need to calculate the coordinates in the active sets.

With the given $\mathbf{E}^t$, updating $\widetilde{\mathbf{x}}_j^{\mathrm{T}}\mathbf{E}^t\mathbf{v}^t$ requires $O(b^t n)$ by only calculating the coordinates corresponding to the active set. Similarly, the complexity of updating $\mathbf{u}^{t\mathrm{T}}\mathbf{X}^{\mathrm{T}}\widetilde{\mathbf{e}}_k^t$ is $O(a^t n)$. The complexity of computing $\|\mathbf{u}^t\|_2^2$ or $\|\mathbf{v}^t\|_2^2$ is $O(a^t)$ or $O(b^t)$, respectively. Comparing (48) with (49), the complexity of (48) is negligible because it only searches j and k in the active sets. By the expansion of loss function, the complexity for comparing the value of loss function is also negligible in terms of (49). Therefore, with $\mathbf{E}^t$, the complexity of backward and forward step in the $(t+1)$th step is $O(a^t n q + b^t n p)$. Combing the complexity for updating $\mathbf{E}^t$, the complexity in the $(t+1)$th step is still of the order $O(a^t n q + b^t n p)$.

D.2 Proof of Lemma 5

Lemma 5 *Let $Q(\cdot)$ be the objective function defined in (7) of the main paper.*
1. For any λ, if there exist j and k such that $Q(h\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}}; \lambda) \leq Q(0; \lambda)$ where $|h| = \epsilon$, it must be true that $\lambda \leq \lambda^0$.
2. For any t, we have $Q(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}; \lambda^{t+1}) \leq Q(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; \lambda^{t+1}) - \xi$.
3. For $\xi \geq 0$ and any t such that $\lambda^{t+1} < \lambda^t$, we have

$$\begin{aligned}
Q(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; \lambda^t) - \xi &\leq Q\left((d^t\mathbf{u}^t \pm \epsilon\mathbf{1}_j)\mathbf{v}^{t\mathrm{T}}; \lambda^t\right),\\
Q(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; \lambda^t) - \xi &\leq Q\left(\mathbf{u}^t(d^t\mathbf{v}^t \pm \epsilon\mathbf{1}_k)^{\mathrm{T}}; \lambda^t\right),
\end{aligned}$$

where $1 \leq j \leq p$ and $1 \leq k \leq q$.

The first statement in Lemma 5 ensures the validity of the initialization step. The second statement in Lemma 5 ensures that, for each λ^t, the algorithm performs coordinate descent whenever $\lambda^t = \lambda^{t+1}$. The third statement implies that λ^t gets reduced only when the penalized loss at λ^t can not be further reduced even by searching over all possible coordinate descent directions.

Proof [Proof of Lemma 5] 1. By the assumption, we have $L(0) - L(h\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}}) \geq \lambda\rho(h\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}}) = \lambda\epsilon$, which yields

$$\lambda \leq \frac{1}{\epsilon}[L(0) - L(h\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}})] \leq \frac{1}{\epsilon}[L(0) - \min_{j,k,s=\pm\epsilon} L(h\mathbf{1}_j\mathbf{1}_k^{\mathrm{T}})] = \lambda^0.$$

2. In the backward step, we have that $\lambda^{t+1} = \lambda^t$, $L(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) \leq L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) + \lambda^t\epsilon - \xi$, and the penalty term is always decreased by a fixed amount $\lambda^t\epsilon$. So $Q(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}; \lambda^{t+1}) \leq Q(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; \lambda^{t+1}) - \xi$ holds.

It remains to consider the forward step when $\lambda^{t+1} = \lambda^t$. If

$$Q(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}; \lambda^{t+1}) > Q(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; \lambda^{t+1}) - \xi,$$

we have

$$L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) - L(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \xi < \lambda^{t+1}[\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1}) - \rho(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}})] = \lambda^{t+1}\epsilon,$$

which yields that

$$\lambda^{t+1} > \frac{L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) - L(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \xi}{\epsilon}.$$

This contradicts with the formula for computing λ^{t+1}.

3. The λ^t can only be reduced in the forward step. So when $\lambda^{t+1} < \lambda^t$, we have

$$\lambda^{t+1} = \frac{L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) - L(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \xi}{\epsilon},$$

and $\epsilon = \rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \rho(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}})$. It follows that

$$\lambda^{t+1}[\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \rho(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}})] = L(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) - L(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \xi.$$

That is, $Q(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}; \lambda^{t+1}) = Q(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; \lambda^{t+1}) - \xi$.

We then have that

$$Q(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}; \lambda^{t+1}) + (\lambda^t - \lambda^{t+1})\rho(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) = Q(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; \lambda^t) - \xi, \tag{50}$$

and

$$\begin{aligned}
&\lambda^{t+1}\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) + (\lambda^t - \lambda^{t+1})\rho(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}) \\
&= \lambda^{t+1}\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) + (\lambda^t - \lambda^{t+1})[\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) - \epsilon] \\
&= \lambda^t\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) + \epsilon(\lambda^{t+1} - \lambda^t) \leq \lambda^t\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}).
\end{aligned} \tag{51}$$

Combining (50) and (51), we have

$$\begin{aligned}
Q(d^t\mathbf{u}^t\mathbf{v}^{t\mathrm{T}}; \lambda^t) - \xi &\leq Q(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}; \lambda^{t+1}) + (\lambda^t - \lambda^{t+1})\rho(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}) \\
&= Q(d^{t+1}\mathbf{u}^{t+1}\mathbf{v}^{t+1\mathrm{T}}; \lambda^t) \\
&= \min\{Q((d^t\mathbf{u}^t \pm \epsilon\mathbf{1}_j)\mathbf{v}^{t\mathrm{T}}; \lambda^t), Q((\mathbf{u}^t(d^t\mathbf{v}^t \pm \epsilon\mathbf{1}_k)^{\mathrm{T}}; \lambda^t)\}.
\end{aligned}$$

This completes the proof. ∎

Now we prove Lemma 5. Recall that for λ^t, the corresponding objective function is

$$Q(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) = L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) + \lambda^t d\|\mathbf{u}\|_1\|\mathbf{v}\|_1,$$

where $L(d\mathbf{u}\mathbf{v}^{\mathrm{T}}) = (2n)^{-1}\|\mathbf{Y} - d\mathbf{X}\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_F^2 + \mu\|d\mathbf{u}\mathbf{v}^{\mathrm{T}}\|_F^2/2$. In the $(t+1)$th step, the objective function is strongly convex when either $\mathbf{u}^t$ or $\mathbf{v}^t$ is fixed. Thus, with any given $\mathbf{X}$ and $\mathbf{Y}$, there exists a constant M such that

$$\mu\|\mathbf{v}^t\|_2^2\mathbf{I} \preceq \nabla_{d\mathbf{u}}^2 L \preceq M\mathbf{I}, \ \mu\|\mathbf{u}^t\|_2^2\mathbf{I} \preceq \nabla_{d\mathbf{v}}^2 L \preceq M\mathbf{I},$$

where $\mathbf{I}$ is the identity matrix, and $\nabla_{d\mathbf{u}}^2 L$ and $\nabla_{d\mathbf{v}}^2 L$ are the second derivatives of the loss function with respect to $d\mathbf{u}$ and $d\mathbf{v}$, respectively. Define

$$(d\mathbf{u})^{t*} = \arg\min_{d\mathbf{u}} Q(d\mathbf{u}\mathbf{v}^{t\mathrm{T}}) \text{ with fixed } \mathbf{v}^t;$$

$$(d\mathbf{v})^{t*} = \arg\min_{d\mathbf{v}} Q(\mathbf{u}^t(d\mathbf{v})^{\mathrm{T}}) \text{ with fixed } \mathbf{u}^t.$$

We now derive the upper bound of $\|d^t\mathbf{u}^t - (d\mathbf{u})^{t*}\|_2$. (With similar argument, we can get the upper bound of $\|d^t\mathbf{v}^t - (d\mathbf{v})^{t*}\|_2$.) With fixed $\mathbf{v}^t$, Taylor expansion of the objective function gives

$$Q\left((d\mathbf{u})^{t*}\right) = Q\left((d\mathbf{u})^t\right) + [\nabla_{d\mathbf{u}}L\left((d\mathbf{u})^t\right) + \lambda^t\boldsymbol{\delta}]^{\mathrm{T}}\left((d\mathbf{u})^{t*} - (d\mathbf{u})^t\right)$$
$$+ \frac{1}{2}\left((d\mathbf{u})^{t*} - (d\mathbf{u})^t\right)^{\mathrm{T}}\nabla_{d\mathbf{u}}^2 L\left((d\mathbf{u})^{t*} - (d\mathbf{u})^t\right), \tag{52}$$

where $\boldsymbol{\delta}$ is a p-dimensional vector and we use $\|\mathbf{v}^t\| = 1$. The jth entry of $\boldsymbol{\delta}$, δ_j, is subject to $|\delta_j| \leq 1$ and $\delta_j = \text{sgn}(u_j)$ if $u_j \neq 0$. Here for convenience, we have rewritten $d^t\mathbf{u}^t$ as $(d\mathbf{u})^t$ and omitted $\mathbf{v}^t$ in the expressions. With the boundedness of the Hessian matrix and the fact that $Q\left((d\mathbf{u})^{t*}\right) \leq Q\left((d\mathbf{u})^t\right)$, it follows that

$$\frac{\mu\|\mathbf{v}^t\|_2^2}{2}\|(d\mathbf{u})^{t*} - (d\mathbf{u})^t\|_2^2 \leq -[\nabla_{d\mathbf{u}}L\left((d\mathbf{u})^t\right) + \lambda^t\boldsymbol{\delta}]^{\mathrm{T}}\left((d\mathbf{u})^{t*} - (d\mathbf{u})^t\right)$$
$$\leq \|\nabla_{d\mathbf{u}}L\left((d\mathbf{u})^t\right) + \lambda^t\boldsymbol{\delta}\|_2\|(d\mathbf{u})^{t*} - (d\mathbf{u})^t\|_2.$$

Therefore,

$$\|(d\mathbf{u})^{t*} - (d\mathbf{u})^t\|_2 \leq \frac{2}{\mu\|\mathbf{v}^t\|_2^2}\|\nabla_{d\mathbf{u}}L\left((d\mathbf{u})^t\right) + \lambda^t\boldsymbol{\delta}\|_2.$$

It remains to derive the upper bound of $\|\nabla_{d\mathbf{u}}L\left((d\mathbf{u})^t\right) + \lambda^t\boldsymbol{\delta}\|_2$. To do this, let's examine the upper bound of each entry of $\nabla_{d\mathbf{u}}L\left((d\mathbf{u})^t\right) + \lambda^t\boldsymbol{\delta}$. By the third statement in Lemma 5, for $u_j^t \neq 0$, we have

$$L\left((d\mathbf{u})^t \pm \text{sgn}(u_j^t)\epsilon\mathbf{1}_j\right) \pm \lambda^t\epsilon \geq L\left((d\mathbf{u})^t\right) - \xi, \tag{53}$$

when $\lambda^{t+1} < \lambda^t$. On the other hand, with Taylor expansion, we have

$$L\left((d\mathbf{u})^t \pm \text{sgn}(u_j^t)\epsilon\mathbf{1}_j\right) = L\left((d\mathbf{u})^t\right) \pm \text{sgn}(u_j^t)\epsilon\nabla_{d\mathbf{u}}L\left((d\mathbf{u})^t\right)^{\mathrm{T}}\mathbf{1}_j$$
$$+ \frac{\epsilon^2}{2}\mathbf{1}_j^{\mathrm{T}}\nabla_{d\mathbf{u}}^2 L\mathbf{1}_j. \tag{54}$$

Combing (53) and (54) yields

$$\mp \left[\mathrm{sgn}(u_j^t)\epsilon \nabla_{d\mathbf{u}} L\left((d\mathbf{u})^t\right)^{\mathrm{T}} \mathbf{1}_j + \lambda^t \epsilon \right] \leq \frac{\epsilon^2}{2} \mathbf{1}_j^{\mathrm{T}} \nabla_{d\mathbf{u}}^2 L \mathbf{1}_j + \xi \leq \frac{M\epsilon^2}{2} + \xi,$$

where the last inequality is due to the boundedness of the Hessian matrix. We have that

$$\begin{aligned}
|\nabla_{d\mathbf{u}} L\left((d\mathbf{u})^t\right)^{\mathrm{T}} \mathbf{1}_j + \mathrm{sgn}(u_j^t)\lambda^t| &= |\mathrm{sgn}(u_j^t)\nabla_{d\mathbf{u}} L\left((d\mathbf{u})^t\right)^{\mathrm{T}} \mathbf{1}_j + \lambda^t| \\
&\leq \frac{M\epsilon}{2} + \frac{\xi}{\epsilon} \quad \text{for } u_j^t \neq 0.
\end{aligned} \tag{55}$$

When $u_j^t = 0$, by the third statement of Lemma 5, we have that

$$L\left((d\mathbf{u})^t \pm \mathrm{sgn}(u_j^t)\epsilon \mathbf{1}_j\right) + \lambda^t \epsilon \geq L\left((d\mathbf{u})^t\right) - \xi. \tag{56}$$

Similarly, combining (54) and (56), we have that

$$\begin{aligned}
\mp \mathrm{sgn}(u_j^t)\epsilon \nabla_{d\mathbf{u}} L\left((d\mathbf{u})^t\right)^{\mathrm{T}} \mathbf{1}_j &\leq \xi + \lambda^t \epsilon + \frac{\epsilon^2}{2} \mathbf{1}_j^{\mathrm{T}} \nabla_{d\mathbf{u}}^2 L \mathbf{1}_j \\
&\leq \xi + \lambda^t \epsilon + \frac{M\epsilon^2}{2},
\end{aligned}$$

which yields

$$|\nabla_{d\mathbf{u}} L\left((d\mathbf{u})^t\right)^{\mathrm{T}} \mathbf{1}_j| - \lambda^t \leq \frac{M\epsilon}{2} + \frac{\xi}{\epsilon} \quad \text{for } u_j^t = 0. \tag{57}$$

Then, combining (55) and (57), we get

$$|\nabla_{d\mathbf{u}} L\left((d\mathbf{u})^t\right)^{\mathrm{T}} \mathbf{1}_j + \delta_j \lambda^t| \leq \frac{M\epsilon}{2} + \frac{\xi}{\epsilon} \quad \text{for } j = 1, \ldots, p.$$

Note that we can choose δ_j appropriately from -1 to 1 when $u_j^t = 0$. Therefore, we get the bound that

$$\begin{aligned}
\|(d\mathbf{u})^{t*} - (d\mathbf{u})^t\|_2 &\leq \frac{2}{\mu\|\mathbf{v}^t\|_2^2} \|\nabla_{d\mathbf{u}} L\left((d\mathbf{u})^t\right) + \lambda^t \boldsymbol{\delta}\|_2 \\
&\leq \frac{2\sqrt{p}}{\|\mathbf{v}^t\|_2^2} \left(\frac{M\epsilon}{2\mu} + \frac{\xi}{\epsilon\mu} \right).
\end{aligned} \tag{58}$$

By similar argument, we can get the bound that

$$\|(d\mathbf{v})^{t*} - (d\mathbf{v})^t\|_2 \leq \frac{2\sqrt{q}}{\|\mathbf{u}^t\|_2^2} \left(\frac{M\epsilon}{2\mu} + \frac{\xi}{\epsilon\mu} \right). \tag{59}$$

Then, with $1 = \|\mathbf{u}^t\|_1 \leq \sqrt{p}\|\mathbf{u}^t\|_2$ and $1 = \|\mathbf{v}^t\|_1 \leq \sqrt{q}\|\mathbf{v}^t\|_2$, combining (58) and (59) yields

$$\max \left[\|(d\mathbf{u})^{t*}\mathbf{v}^{t\mathrm{T}} - d^t \mathbf{u}^t \mathbf{v}^{t\mathrm{T}}\|_F, \ \|\mathbf{u}^t(d\mathbf{v})^{t*\mathrm{T}} - d^t \mathbf{u}^t \mathbf{v}^{t\mathrm{T}}\|_F \right] \leq 2\sqrt{pq} \left(\frac{M\epsilon}{2\mu} + \frac{\xi}{\epsilon\mu} \right),$$

where we make use of the fact that $\|\mathbf{x}\mathbf{y}^{\mathrm{T}}\|_F = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$ for any column vectors $\mathbf{x}$ and $\mathbf{y}$. This completes the proof.

References

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

Mihye Ahn, Haipeng Shen, Weili Lin, and Hongtu Zhu. A sparse reduced rank framework for group analysis of functional neuroimaging data. *Statistica Sinica*, 25(1):295–312, 2015.

Genevera I. Allen, Logan Grosenick, and Jonathan Taylor. A generalized least-square matrix decomposition. *Journal of the American Statistical Association*, 109(505):145–159, 2014.

T. W. Anderson. Estimating linear restrictions on regression coefficients for multivariate normal distributions. *Annals of Mathematical Statistics*, 22:327–351, 1951.

Marie Z. Bao, Monica A. Schwartz, Greg T. Cantin, John R. Yates III, and Hiten D. Madhani. Pheromone-dependent destruction of the Tec1 transcription factor is required for MAP kinase signaling specificity in yeast. *Cell*, 119(7):991–1000, 2004.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

Rachel B. Brem and Leonid Kruglyak. The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences*, 102(5):1572–1577, 2005.

Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T. Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, Adrian Cortes, Samantha Welsh, Alan Young, Mark Effingham, Gil McVean, Stephen Leslie, Naomi Allen, Peter Donnelly, and Jonathan Marchini. The UK biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209, 2018.

Emmanuel J. Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98 (6):925–936, 2010.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, December 2009.

Kun Chen, Kung-Sik Chan, and Nils Chr. Stenseth. Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B*, 74(2): 203–221, 2012.

Kun Chen, Kung-Sik Chan, and Nils Chr. Stenseth. Source-sink reconstruction through regularized multicomponent regression analysis–with application to assessing whether North Sea cod larvae contributed to local fjord cod in Skagerrak. *Journal of the American Statistical Association*, 109 (506):560–573, 2014.

Lisha Chen and Jianhua Z. Huang. Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, 107(500): 1533–1545, 2012.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert J. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Yingying Fan and Jinchi Lv. Asymptotic properties for combined ℓ_1 and concave regularization. *Biometrika*, 101(1):57–70, 2014.

Yingying Fan and Cheng Yong Tang. Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society: Series B*, 75(3):531–552, 2013.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.

Michael C. Gustin, Jacobus Albertyn, Matthew Alexander, and Kenneth Davenport. MAP kinase pathways in the yeast saccharomyces cerevisiae. *Microbiology and Molecular Biology Reviews*, 62(4):1264–1300, 1998.

Trevor Hastie, Robert J. Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2008.

Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. *Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

Lifang He, Kun Chen, Wanwan Xu, Jiayu Zhou, and Fei Wang. Boosted sparse and low-rank tensor regression. In *Advances in Neural Information Processing Systems (NeurIPS) 31*, pages 1009–1018. Curran Associates, Inc., 2018.

Minoru Kanehisa, Susumu Goto, Miho Furumichi, Mao Tanabe, and Mika Hirakawa. KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Research*, 38:D355–D360, 2009.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

Mihee Lee, Haipeng Shen, Jianhua Z. Huang, and James S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, 2010.

Xin Ma, Luo Xiao, and Wing Hung Wong. Learning regulatory programs by threshold SVD regression. *Proceedings of the National Academy of Sciences*, 111(44):15675–15680, 2014.

Zhuang Ma, Zongming Ma, and Tingni Sun. Adaptive estimation in two-way sparse reduced-rank regression. *Statistica Sinica*, 30:2179–2201, 2020.

Arin Minasian, Shahram ShahbazPanahi, and Raviraj S. Adve. Energy harvesting cooperative communication systems. *IEEE Transactions on Wireless Communications*, 13(11):6118–6131, 2014.

Aditya Mishra, Dipak K. Dey, and Kun Chen. Sequential co-sparse factor regression. *Journal of Computational and Graphical Statistics*, 26(4):814–825, 2017.

Junyang Qian, Yosuke Tanigawa, Ruilin Li, Robert J. Tibshirani, Manuel A. Rivas, and Trevor Hastie. Large-scale sparse regression for multiple responses with applications to UK biobank. *Annals of Applied Statistics*, 16(3):1891–1918, 2022.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL https://www.R-project.org/.

Gregory C. Reinsel and Raja P. Velu. *Multivariate Reduced-rank Regression: Theory and Applications*. New York: Springer, 1998.

John D. Storey, Joshua M. Akey, and Leonid Kruglyak. Multiple locus linkage analysis of genomewide expression in yeast. *PLoS biology*, 3(8):e267, 2005.

Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. SVDNet for pedestrian retrieval. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3820–3828, 2017.

Ryan J. Tibshirani. A general framework for fast stagewise algorithms. *Journal of Machine Learning Research*, 16:2543–2588, 2015.

Yoshimasa Uematsu, Yingying Fan, Kun Chen, Jinchi Lv, and Wei Lin. SOFAR: Large-scale association network learning. *IEEE Transactions on Information Theory*, 65(8):4924–4939, 2019.

Gregory Vaughan, Robert Aseltine, Kun Chen, and Jun Yan. Stagewise generalized estimation equations with grouped variables. *Biometrics*, 73(4):1332–1342, 2017.

Maria Vounou, Thomas E. Nichols, and Giovanni Montana. Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach. *NeuroImage*, 53(3):1147–1159, 2010.

Daniela M. Witten, Robert J. Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

Dan Yang, Zongming Ma, and Andreas Buja. A sparse singular value decomposition method for high-dimensional data. *Journal of Computational and Graphical Statistics*, 23(4):923–942, 2014.

Fei Ye and Cun-Hui Zhang. Rate minimaxity of the Lasso and Dantzig selector for the l_q loss in l_r balls. *Journal of Machine Learning Research*, 11:3519–3540, 2010.

Ming Yu, Varun Gupta, and Mladen Kolar. Recovery of simultaneous low rank and two-way sparse coefficient matrices, a nonconvex approach. *Electronic Journal of Statistics*, 14(1):413–457, 2020.

Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2014.

Cun-Hui Zhang and Tong Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593, 2012.

Peng Zhao and Bing Yu. Stagewise Lasso. *Journal of Machine Learning Research*, 100(8):2701–2726, 2013.

Zemin Zheng, M. Taha Bahadori, Yan Liu, and Jinchi Lv. Scalable interpretable multi-response regression via SEED. *Journal of Machine Learning Research*, 20(107):1–34, 2019.

Zemin Zheng, Jinchi Lv, and Wei Lin. Nonsparse learning with latent variables. *Operations Research*, 69(1):346–359, 2021.

Zemin Zheng, Yang Li, Jie Wu, and Yuchen Wang. Sequential scaled sparse factor regression. *Journal of Business and Economic Statistics*, 40(2):595–604, 2022.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.