Structural pursuit over multiple undirected graphs *

Yunzhang Zhu[1], Xiaotong Shen[1] and Wei Pan[2]

Summary

Gaussian graphical models are useful to analyzing and visualizing conditional dependence relationships between interacting units. Motivated from network analysis under different experimental conditions, such as gene networks for disparate cancer subtypes, we model structural changes over multiple networks with possible heterogeneities. In particular, we estimate multiple precision matrices describing dependencies among interacting units through maximum penalized likelihood. Of particular interest are homogeneous groups of similar entries across and zero-entries of these matrices, referred to as clustering and sparseness structures, respectively. A non-convex method is proposed to seek a sparse representation for each matrix and identify clusters of the entries across the matrices. Computationally, we develop an efficient method on the basis of difference convex programming, the augmented Lagrangian method and the block-wise coordinate descent method, which is scalable to hundreds of graphs of thousands nodes through a simple necessary and sufficient partition rule, which divides nodes into smaller disjoint subproblems excluding zero-coefficients nodes for arbitrary graphs with convex relaxation. Theoretically, a finite-sample error bound is derived for the proposed method to reconstruct the clustering and sparseness structures. This leads to consistent reconstruction of these two structures simultaneously, permitting the number of unknown parameters to be exponential in the sample size, and yielding the optimal performance of the oracle estimator as if the true structures were given *a priori*. Simulation studies suggest that the method enjoys the benefit of pursuing these two disparate kinds of structures, and compares favorably against its convex counterpart in the accuracy of structure pursuit and parameter estimation.

Key Words: Simultaneous pursuit of sparseness and clustering, multiple networks, non-convex, prediction, signaling network inference.

1 Introduction

Graphical models are widely used to describe relationships among interacting units. Major components of the models are nodes that represent random variables, and edges encoding conditional dependencies between the nodes. Of great current interest is the identification of certain lower-dimensional structures for undirected graphs. The central topic of this

*[1]School of Statistics, [2]Division of Biostatistics, University of Minnesota, Minneapolis, MN 55455. This research was supported in part by the National Science Foundation Grant DMS-1207771 and National Institutes of Health Grants R01GM081535, HL65462 and R01HL105397. The authors thank the editors and the reviewers for helpful comments and suggestions.

paper is maximum penalized likelihood estimation of multiple Gaussian graphical models for simultaneously pursuing two disparate kinds of structures–sparseness and clustering.

In the literature on Gaussian graphical models, the current research effort has concentrated on reconstruction of a *single* sparse graph. Methods to exploit matrix sparsity include [1, 5, 11, 13, 14, 15, 24], among others. For *multiple* Gaussian graphical models, existing approaches mainly focus on either exploring temporal smoothing structure [7, 25] or encouraging common sparsity across the networks [6, 10]. In this paper, we focus on pursuing both clustering and sparseness structures over multiple graphs, including temporal clustering as a special case while allowing for abrupt changes of structures over graphs. For multiple graphs without a temporal ordering, our method enables to identify possible element-wise heterogeneity among undirected graphs. This is motivated by heterogeneous gene regulatory networks corresponding to disparate cancer subtypes [18, 21]. In such a situation, the overall associations among genes remain similar for each network, whereas specific pathways and certain critical nodes (genes) may be differentiated under disparate conditions.

For multiple Gaussian graphical models, estimation is challenging due to enormous candidate graphs of order 2^{Lp^2}, where p is the total number of nodes and L is total number of graphs. To battle the curse of dimensionality, we explore two dissimilar types of structures simultaneously: (1) sparseness within each graph and (2) element-wise clustering across graphs. The benefit of exploration is three-fold. First, it goes beyond sparseness pursuit alone for each graph, which is usually inadequate given a large number of unknown parameters relative to the sample size, as demonstrated in four numerical examples in Section 5. Second, borrowing information across graphs enables us to detect the changes of sparseness and clustering structures over the multiple graphs. Third, pursuit of these two structures at the same time is suited for our problem, which seeks both similarities and differences among the multiple graphs.

To this end, we propose a regularized/constrained maximum likelihood method for si-

multaneous pursuit of sparseness and clustering structures. Computationally, we develop a strategy to convert the optimization involving matrices to a sequence of much simpler quadratic problems to solve. Most critically, we derive a necessary and sufficient partition rule to partition the nodes into disjoint subproblems excluding zero-coefficient nodes for multiple arbitrary graphs with convex relaxation, where the rule is applied before computation is performed. Such a rule has been used in [12] for convex estimation of a single matrix, but has not been available for multiple arbitrary graphs, to our knowledge. This makes efficient computation possible for multiple large graphical models, which otherwise is rather difficult if not impossible. Theoretically, we develop a novel theory for the proposed method, and show that it enables to reconstruct the oracle estimator as if the true sparseness and element-wise clustering structures were given *a priori*, which leads to reconstruction of the two types of structures consistently. This occurs roughly when the size of L matrices p^2L is of order $\exp(An)$, where p is the dimension of the matrices and A is related to the Hessian matrices of the negative log-determinant of the true precision matrices and the resolution level for simultaneous pursuit of sparseness and element-wise clustering, c.f., Corollary 2. Moreover, we quantify the degree of improvement due to structural pursuit beyond that of sparsity.

The rest of this article is organized as follows. Section 2 introduces the proposed method. Section 3 is devoted to estimation of partial correlations across multiple graphical models, and develops computational tools for efficient computation. Section 4 presents a theory concerning the accuracy of structural pursuit and parameter estimation, followed by some numerical examples in Section 5 and an application to signaling network inference in Section 6. Section 7 discusses various issues in modeling. Finally, the appendix contains proofs.

2 Proposed method

Consider the $L-$sample problem with the $l-$th sample $\boldsymbol{X}_1^{(l)}, \cdots, \boldsymbol{X}_{n_l}^{(l)}$ from $\mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$; $l = 1, \cdots, L$, we estimate $\boldsymbol{\Omega} = (\boldsymbol{\Omega}_1, \cdots, \boldsymbol{\Omega}_L)$, where $\boldsymbol{\Omega}_l = \boldsymbol{\Sigma}_l^{-1}$ is the $p \times p$ inverse covariance matrix and positive definite, denoted by $\boldsymbol{\Omega}_l \succ 0$, $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ are the corresponding mean vector and covariance matrix, and the sample size $n = \sum_{l=1}^{L} n_l$.

For maximum likelihood estimation, the profile likelihood for $\boldsymbol{\Omega}$, after $\mu_1, \cdots, \mu_L$ are maximized out, is proportional to

$$\sum_{l=1}^{L} n_l \big(\log\det(\boldsymbol{\Omega}_l) - tr(\boldsymbol{S}_l \boldsymbol{\Omega}_l) \big), \tag{1}$$

where $\bar{\boldsymbol{X}}_l = n_l^{-1} \sum_{i=1}^{n_l} \boldsymbol{X}_i^{(l)}$ and $\boldsymbol{S}_l = n_l^{-1} \sum_{i=1}^{n_l} (\boldsymbol{X}_i^{(l)} - \bar{\boldsymbol{X}}_l)(\boldsymbol{X}_i^{(l)} - \bar{\boldsymbol{X}}_l)^T$ are the corresponding sample mean and covariance matrix, det and tr denote the determinant and trace. In (1), the number of unknown parameters in $\boldsymbol{\Omega}$ can greatly exceed the sample size n.

2.1 General penalized multiple precision matrices estimation

To avoid non-identifiability in (1) and encourage low dimensional structures, we propose a regularized maximum likelihood approach through penalty functions $J_{jk}(\cdot)$:

$$\text{maximize}_{\boldsymbol{\Omega} \succ 0} \; S(\boldsymbol{\Omega}) = \sum_{l=1}^{L} n_l \big(\log\det(\boldsymbol{\Omega}_l) - tr(\boldsymbol{S}_l \boldsymbol{\Omega}_l) \big) - \sum_{j \neq k} J_{jk}(\omega_{jk1}, \cdots, \omega_{jkL}), \tag{2}$$

where $\boldsymbol{\Omega} = (\boldsymbol{\Omega}_1, \cdots, \boldsymbol{\Omega}_L)$ and only the off-diagonals $\{\omega_{jkl}\}$ of $\boldsymbol{\Omega}_l$ are regularized. Note that $J_{jk}(\cdot)$ could be any function that penalizes jk-th entries across $\boldsymbol{\Omega}_l$'s. This encompasses many existing penalty-based approaches for multiple Gaussian graphical models [6, 22] as special cases.

In general, the maximization problem (2) involving L matrices is computationally difficult. To meet the computational challenges, we develop a general block-wise coordinate descent strategy to reduce (2) to an iterative procedure involving much easier subproblems. Before proceeding, we introduce some notations. Let the jth row (or column) of $\boldsymbol{\Omega}_l$ be $\boldsymbol{\omega}_{jl}$, let $\boldsymbol{\omega}_{-jl} = (\omega_{j1l}, \ldots, \omega_{j(j-1)l}, \omega_{j(j+1)l}, \ldots, \omega_{jpl})$ be a $(p-1)$-dimensional vector, excluding the

jth component of $\boldsymbol{\omega}_{jl}$, and $\boldsymbol{\Omega}_{-jl}$ be the sub-matrix without the jth row and column of $\boldsymbol{\Omega}_l$, and $\boldsymbol{\Omega}_{-jl}^{-1}$ be the inverse of $\boldsymbol{\Omega}_{-jl}$.

Our proposed method maximizes (2) by sweeping each row (or column) of $\boldsymbol{\Omega}$ across $l = 1, \cdots, L$. Using the property that $\det(\boldsymbol{\Omega}_l) = \det\big(\boldsymbol{\Omega}_{-jl}(\boldsymbol{\omega}_{jjl} - \boldsymbol{\omega}_{-jl}^T \boldsymbol{\Omega}_{-jl}^{-1} \boldsymbol{\omega}_{-jl})\big)$ with ^{T} indicating the transpose, we rewrite (2), after ignoring constant terms, as a function of each row (or column) $(\boldsymbol{\omega}_{j1}, \cdots, \boldsymbol{\omega}_{jl})$ across l; $j = 1, \cdots, p$,

$$\sum_{l=1}^{L} n_l \big(\log\big(\omega_{jjl} - \boldsymbol{\omega}_{-jl}^T \boldsymbol{\Omega}_{-jl}^{-1} \boldsymbol{\omega}_{-jl}\big)\big) - s_{jjl}\omega_{jjl} - 2\boldsymbol{s}_{-jl}^T \boldsymbol{\omega}_{-jl}\big) - \sum_{k \neq j} J_{jk}(\omega_{jk1}, \cdots, \omega_{jkL}). \quad (3)$$

First, for each fixed row (or column) of $\boldsymbol{\Omega}$ across $l = 1, \cdots, L$, we maximize (3) over the diagonals $(\omega_{jj1}, \cdots, \omega_{jjl})$ given the corresponding off-diagonals $(\boldsymbol{\omega}_{-j1}, \cdots, \boldsymbol{\omega}_{-jl})$. Setting the partial derivatives of (3) in the diagonals to be zero yields the profile maximizer of (2)

$$\hat{\omega}_{jjl} \;=\; 1/s_{jjl} + \boldsymbol{\omega}_{-jl}^T \boldsymbol{\Omega}_{-jl}^{-1} \boldsymbol{\omega}_{-jl}, \quad l = 1, \cdots, L. \quad (4)$$

Second, substituting (4) into (3) yields the negative profile likelihood of (2) for $(\boldsymbol{\omega}_{-jl}, \cdots, \boldsymbol{\omega}_{-jl})$

$$\sum_{l=1}^{L} n_l \big(s_{jjl} \boldsymbol{\omega}_{-jl}^T \boldsymbol{\Omega}_{-jl}^{-1} \boldsymbol{\omega}_{-jl} + 2\boldsymbol{s}_{-jl}^T \boldsymbol{\omega}_{-jl}\big) + \sum_{k \neq j} J_{jk}(\omega_{jk1}, \cdots, \omega_{jkL}). \quad (5)$$

Third, the aforementioned process is repeated for each rows (or columns) of $\boldsymbol{\Omega}$ until a certain stopping criterion is satisfied. By Theorem 1, profiling is equivalent to the original problem for separable convex penalty functions summarized as follows.

Theorem 1 *Iteratively minimizing* (5) *over the off-diagonals* $(\boldsymbol{\omega}_{-j1}, \cdots, \boldsymbol{\omega}_{-jL})$ *and updating diagonals* ω_{jjl} *by* (4)*;* $j = 1, \cdots, p, l = 1, \cdots, L$ *converges to a local maximizer of* (2)*. Moreover, if* $J_{jk}(\cdot)$ *are convex, it converges to a global maximizer.*

Theorem 1 reduces (2) to iteratively solving (5) that is quadratic in its argument. On this ground we design efficient methods for solving (2) with a specific choice of $J_{jk}(\cdot)$ next.

2.2 Pursuit of sparseness and clustering structures

A zero element in $\boldsymbol{\Omega}_l$ corresponds to conditional independence between two components of $Y^{(l)}$ given its other components [9]. Thus, within each precision matrix $\boldsymbol{\Omega}_l$, estimating its

elements reconstructs its graph structure, where a zero-element of $\mathbf{\Omega}_l$ corresponds to no edges between the two nodes, encoding conditional independence. In addition, the nodes connecting many other nodes are identified, called network hubs. On the other hand, over multiple precision matrices, estimating element-wise clustering structure can reveal the change of sparseness and clustering structures.

To detect clustering structures, consider element-wise clustering of entries of $\mathbf{\Omega}_1, \cdots, \mathbf{\Omega}_L$ based on possible prior knowledge. The prior knowledge is specified loosely in an undirected graph $\mathcal{U}$ with each node corresponding to a triplet (j,k,l); $1 \le j < k \le p$, $1 \le l \le L$. That is, an edge between node (j,k,l) and (j,k,l') means that the (j,k)th entry of $\mathbf{\Omega}_l$ and the (j,k)th entry of $\mathbf{\Omega}_{l'}$ tend to be similar *a priori* and thus can be pushed to share the same value. Specifically, let $\mathcal{E}_{jk}$ denote a set of edges between two distinct nodes $(j,k,l) \neq (j,k,l')$ of $\mathcal{U}$, where $(l,l') \in \mathcal{E}_{jk}$ indicates a connection between the two nodes $(j,k,l), (j,k,l')$. To identify homogeneous subgroups of off-diagonals $\{\omega_{jkl}\}$ of $\mathbf{\Omega}_l$ across $l = 1, \cdots, L$ over $\mathcal{U}$, including the group of zero-elements, we propose a non-convex penalty of the form

$$J_{jk}(\omega_{jk1}, \cdots, \omega_{jkL}) = \lambda_1 \sum_{l=1}^{L} J_\tau(|\omega_{jkl}|) + \lambda_2 \sum_{(l,l') \in \mathcal{E}_{jk}} J_\tau(|\omega_{jkl} - \omega_{jkl'}|), \tag{6}$$

to regularize (2), where λ_1 and λ_2 are nonnegative tuning parameters controlling the degrees of sparseness and clustering, $J_\tau(z) = \min(|z|, \tau)$ is the truncated L_1-penalty of [17], called TLP in what follows, which, after rescaled by $\frac{1}{\tau}$, approximates the L_0-function when tuning parameter $\tau > 0$ tends to 0^+.

Note that our approach is applicable to a variety of applications by specifying the graph $\mathcal{U}$. For time varying graphs, our method can be used to detect the change of clustering structure, where $\mathcal{E}_{jk}$ is a serial graph as in the fused Lasso [19], and a serial temporal relation is defined only for elements in adjacent matrices. One key difference between our method and the smoothing method [25, 7] is that it enables to accommodate abrupt changes of structures over networks. For multiple graphs without a serial ordering, the proposed method enables to identify possible element-wise heterogeneity among undirected graphs, such as

gene regulatory networks corresponding to disparate cancer subtypes [18, 21]. Heterogeneity of this type can be dealt with by specifying a complete graph for each $\mathcal{E}_{jk}$.

3 Computation

This section proposes a relaxation method to treat non-convex penalties in (6). For large-scale problems, a partition rule may be useful, which breaks large matrices into many small ones to process separately. A novel necessary and sufficient partition rule is derived for our non-convex penalization method as well as its convex counterpart, generalizing the results for single precision matrix estimation [12, 22].

3.1 Non-convex optimization

For the non-convex minimization (2) with (6), we develop a relaxation method by solving a sequence of convex problems. This method integrates difference convex (DC) programming with block-wise coordinate descent method based on the foregoing strategy.

For DC programming, we first decompose $S(\mathbf{\Omega})$ into a difference of two convex functions: $S(\mathbf{\Omega}) = S_1(\mathbf{\Omega}) - S_2(\mathbf{\Omega})$, with

$$\begin{aligned} S_1(\mathbf{\Omega}) &= \sum_{l=1}^{L} n_l\big(\log\det(\mathbf{\Omega}_l) - tr(\boldsymbol{S}_l\mathbf{\Omega}_l)\big) + \lambda_1 \sum_{(j,k,l):j\neq k} |\omega_{jkl}| + \lambda_2 \sum_{1\le j\neq k\le p} \sum_{(l,l')\in\mathcal{E}_{jk}} |\omega_{jkl} - \omega_{jk'l'}|, \\ S_2(\mathbf{\Omega}) &= \sum_{j\neq k} \lambda_1 \sum_{l=1}^{L} \max(|\omega_{jkl}| - \tau, 0) + \lambda_2 \sum_{1\le j\neq k\le p} \sum_{(l,l')\in\mathcal{E}_{jk}} \max(|\omega_{jkl} - \omega_{jk'l'}| - \tau, 0), \end{aligned} \tag{7}$$

where a DC decomposition of $J_\tau(|z|) = |z| - \max(|z| - \tau, 0)$ is used. Then the trailing convex function $S_2(\mathbf{\Omega})$ is iteratively approximated by its minorization, say at iteration m, $\lambda_1 \sum_{l=1}^{L} \sum_{j\neq k} (\mathbb{I}(|\hat{\omega}_{jkl}^{(m)}| \le \tau)|\omega_{jkl}| + \lambda_2 \sum_{1\le j\neq k\le p} \sum_{(l,l')\in\mathcal{E}} \mathbb{I}(|\hat{\omega}_{jkl}^{(m)} - \hat{\omega}_{jkl'}^{(m)}| \le \tau)|\omega_{jkl} - \omega_{jkl'}|$. This is obtained through minorization $|z^{(m)}| + \zeta(|z^{(m)}|)(|z| - |z^{(m)}|$ of $\max(|z| - \tau, 0)$ at $|\hat{\omega}_{jkl}^{(m)}|$, which is the solution at iteration $m-1$, where $\zeta(|z^{(m)}|)$ is the gradient of $\max(|z| - \tau, 0)$ at $|z^{(m)}|$; see [17] for more discussions about minorization of this type. At iteration m, the cost

function to minimize is

$$-\sum_{l=1}^{L} n_l\Big(\log\det(\boldsymbol{\Omega}_l) - tr(\boldsymbol{S}_l\boldsymbol{\Omega}_l)\Big) + \lambda_1 \sum_{(j,k,l)\in E^{(m)}} |\omega_{jkl}| + \lambda_2 \sum_{\{(j,k,l),(j,k,l')\}\in F^{(m)}} |\omega_{jkl} - \omega_{jkl'}| \quad (8)$$

subject to $\boldsymbol{\Omega}_l \succ 0$; $l = 1, \cdots, L$, where $E^{(m)} = \{(j,k,l) : |\hat{\omega}^{(m)}_{jkl}| \leq \tau, j \neq k\}$; $F^{(m)} = \{\{(j,k,l),(j,k,l')\} : (l,l') \in \mathcal{E}_{jk}, |\hat{\omega}^{(m)}_{jkl} - \hat{\omega}^{(m)}_{jkl'}| \leq \tau\}$.

To solve (8), we apply Theorem 1 to iteratively minimize:

$$\begin{aligned}\sum_{l=1}^{L} n_l\big(s_{jjl}\boldsymbol{\omega}^T_{-jl}\boldsymbol{\Omega}^{-1}_{-jl}\boldsymbol{\omega}_{-jl} + 2\boldsymbol{s}^T_{-jl}\boldsymbol{\omega}_{-jl}\big) \quad &+ \quad \lambda_1 \sum_{(j,k,l)\in E^{(m)}} |\omega_{jkl}| \\ &+ \quad \lambda_2 \sum_{\{(j,k,l),(j,k,l')\}\in F^{(m)}} |\omega_{jkl} - \omega_{jkl'}|, \quad (9)\end{aligned}$$

and update diagonal elements using (4). This quadratic problem can then be efficiently solved using augmented Lagrangian methods as in [26].

Unlike the coordinate descent method updating one component at a time, we update one component of $\boldsymbol{\zeta}$ and two components $(\omega_{jkl}, \omega_{jjl})$ for $\boldsymbol{\Omega}_l$ at the same time.

In (9), computation of $\boldsymbol{\Omega}^{-1}_{-jl}$ by directly inverting $\boldsymbol{\Omega}_{-jl}$ has a complexity of $O(p^3)$ operations for each (j,l). For efficient computation, we utilize the special property of our sweeping operator in that the $(p-1)^2$ elements of $\boldsymbol{\Omega}_l$ are unchanged except one row and one column are swept, in addition to the rank one property for updating the formula. In (9), we derive an analytic formula through block-wise inversion and the Neumann formula of a square matrix, to compute $(\boldsymbol{\Omega}_{-jl})^{-1}$ from $(\omega_{jjl}, \boldsymbol{\omega}_{-jl}, \boldsymbol{\Omega}^{-1}_l)$ and $\boldsymbol{\Omega}^{-1}_l$ from $(\omega_{jjl}, \boldsymbol{\omega}_{-jl}, (\boldsymbol{\Omega}^{-1}_{-l})_{-j})$ for each (j,l). That is,

$$(\boldsymbol{\Omega}_{-jl})^{-1} = (\boldsymbol{\Omega}^{-1}_l)_{-j} - \tfrac{(\boldsymbol{\Omega}^{-1}_l)_j(\boldsymbol{\Omega}^{-1}_l)^T_j}{(\boldsymbol{\Omega}^{-1}_l)_{jj}}, \quad (10)$$

$$\boldsymbol{\Omega}^{-1}_l = \begin{pmatrix} (\boldsymbol{\Omega}^{-1}_l)_{-j} + b\boldsymbol{a}\boldsymbol{a}^T & -b\boldsymbol{a} \\ -b\boldsymbol{a}^T & b \end{pmatrix}, \boldsymbol{a} = (\boldsymbol{\Omega}^{-1}_l)_{-j}\boldsymbol{\omega}_{-jl}, b = (\omega_{jjl} - \boldsymbol{a}^T\boldsymbol{\omega}_{-jl})^{-1}. \quad (11)$$

This amounts to $O(p^2)$ operations.

The foregoing discussion leads to our DC block-wise coordinate descent algorithm through sweeping operations over $p(p-1)$ off-diagonals of $(\boldsymbol{\Omega}_1, \cdots, \boldsymbol{\Omega}_L)$, with each operation involving

the L corresponding off-diagonals.

Algorithm 1:

Step 1. (Initialization) Set $\hat{\mathbf{\Omega}}_l^{(0)} = I$; $l = 1, \cdots, L$, $E^{(0)} = \{(j,k,l) : 1 \leq j \neq k \leq p, 1 \leq l \leq L\}$, $F^{(0)} = \mathcal{E}$, $m = 0$ and precision tolerance $\epsilon = 10^{-5}$ for **Step 2**.

Step 2. (Iteration) At current iteration m, initialize $\mathbf{\Omega} = \hat{\mathbf{\Omega}}^{(m)}$. Then solve (8) applying the block-wise coordinate descent algorithm to update $\mathbf{\Omega}$ to yield $\hat{\mathbf{\Omega}}^{(m+1)}$. And set $E^{(m+1)} = \{(j,k,l) : |\hat{\omega}_{jkl}^{(m+1)}| \leq \tau, j \neq k\}$; $F^{(m+1)} = \{\{(j,k,l),(j',k',l')\} : (l,l') \in \mathcal{E}_{jk}, |\hat{\omega}_{jkl}^{(m+1)} - \hat{\omega}_{j'k'l'}^{(m+1)}| \leq \tau\}$. Specifically,

a) For each row (column) index $j = 1, \cdots, p$, compute $\mathbf{\Omega}_{-jl}^{-1}$ using (10); $l = 1, \cdots, L$. Solve (9) to obtain $\hat{\boldsymbol{\omega}}_{-jl}^{(m)}$, and then compute $\hat{\omega}_{jjl}^{(m)}$ through (4); $l = 1, \cdots, L$. Update $\mathbf{\Omega}_l^{(m)}$ with its jth row replaced by $(\hat{\omega}_{jjl}^{(m)}, \hat{\boldsymbol{\omega}}_{-jl}^{(m)})$ and its jth column by symmetry. Finally update $(\mathbf{\Omega}_l^{(m)})^{-1}$ using (11). Go to next iteration $j+1$ until all rows of $\mathbf{\Omega}_l^{(m)}$ have been swept.

b) Repeat a) until the decrement of the objective function is less than ϵ. After convergence, update $\mathbf{\Omega}$ to yield $\hat{\mathbf{\Omega}}^{(m+1)} = \mathbf{\Omega}$ based on a).

Step 3. (Stopping criterion) Terminate when $E^{(m+1)} = E^{(m)}$ and $F^{(m+1)} = F^{(m)}$, otherwise, repeat **Step 2** with $m = m + 1$.

The overall complexity of Algorithm 1 is of order $O(p^3L^2)$. And real computational time of our algorithm depends highly on values of λ_1, λ_2 and the number of iterations. In Example 1, it takes about 30 seconds for one simulation run with $(p, L) = (200, 4)$ over 100 grids on a 8-core computer with Intel(R) Core(TM) i7-3770 processors and 16GB of RAM.

3.2 Partition rule for large-scale problems

This section establishes a necessary and sufficient partition rule for our non-convex penalization method and its convex counterpart using the sample covariances, permitting fast computation for large-scale problems by partitioning nodes into disjoint subsets excluding the zero-coefficient subset then applying the proposed method to each nonzero subset. Such

a result exists only for a single matrix or a special case of multiple matrices, c.f., [12, 22].

In what follows, we only consider the case where $\mathcal{E}_{jk} = \mathcal{E}$ are identical. Given this graph $\mathcal{G} = (V, \mathcal{E})$, with $(V = \{1, \cdots, L\}, \mathcal{E} = \mathcal{E}_{jk}, 1 \le j < k \le p)$ denoting the node and edge sets, we write $l \sim l'$ if $(l, l') \in \mathcal{E}$, or two nodes are connected. First consider the convex grouping penalty over $\mathcal{G}$, followed by a general case, where the penalized log-likelihood is

$$\sum_{l=1}^{L} \Big(n_l \big(-\log\det(\mathbf{\Omega}_l) + tr(\mathbf{\Omega}_l \boldsymbol{S}_l) \big) + \lambda_1 \|\mathbf{\Omega}_{l,\text{off}}\|_1 \Big) + \lambda_2 \sum_{l \sim l'} \|\mathbf{\Omega}_{l,\text{off}} - \mathbf{\Omega}_{l',\text{off}}\|_1, \quad (12)$$

where $\mathbf{\Omega}_{l,\text{off}}$ denotes the off-diagonal elements of $\mathbf{\Omega}_l$ and $\boldsymbol{S}_l = (s_{jkl})_{1 \le j,k \le p}$ are the sample covariance matrices, $l = 1, \cdots, L$.

The next theorem derives a necessary and sufficient condition for the jkth element of $\hat{\mathbf{\Omega}}_l$ $\hat{\Omega}_{jkl} = 0$ across $l = 1, \cdots, L$, for $j \in \mathcal{J}$, $k \in \mathcal{J}^c$, where $(\hat{\mathbf{\Omega}}_1, \cdots, \hat{\mathbf{\Omega}}_L)$ is the minimizer of (12), and $\mathcal{J} \subset \{1, \cdots, p\}$ is any subset. This partitions the node set into disjoint subsets of connected nodes, with no connections between these subsets.

Theorem 2 *(Partition rule for (12))* $\hat{\Omega}_{jkl} = 0$ *for all* $j \in \mathcal{J}$; $k \in \mathcal{J}^c$ *and* $l = 1, \cdots, L$, *if and only if* $(s_{jk1}, \cdots, s_{jkL}) \in \mathcal{S}$, *for all* $j \in \mathcal{J}, k \in \mathcal{J}^c$, *where* $\mathcal{S} = \big\{ \boldsymbol{s} = (s_1, \cdots, s_L) : |\sum_{l \in \mathcal{I}} n_l s_l| \le \lambda_1 |\mathcal{I}| + \lambda_2 d(\mathcal{I}, \mathcal{I}^c), \forall \mathcal{I} \subseteq V \big\}$ *with* $d(\mathcal{I}, \mathcal{I}^c) = \sum_{l \in \mathcal{I}, l' \in \mathcal{I}^c} \mathbb{I}\big(l \sim l'\big)$ *denoting the number of edges between the nodes in* $\mathcal{I}$ *and the remaining nodes in* $\mathcal{I}^c$.

Similar results hold for the proposed non-convex regularized estimators.

Theorem 3 *(Partition rule for non-convex regularization) Denote by* $\hat{\mathbf{\Omega}}^{dc}$ *the solution obtained from* **Algorithm 1** *for (2). Similarly, given any* $\mathcal{J}$, $\hat{\Omega}^{dc}_{jkl} = 0$ *for all* $j \in \mathcal{J}$; $k \in \mathcal{J}^c$; $l = 1, \cdots, L$, *if and only if* $(s_{jk1}, \cdots, s_{jkL}) \in \mathcal{S}$, *where* $\mathcal{S} = \big\{ \boldsymbol{s} = (s_1, \cdots, s_L) : |\sum_{l \in \mathcal{I}} n_l s_l| \le \lambda_1 |\mathcal{I}| + \lambda_2 d(\mathcal{I}, \mathcal{I}^c), \forall \mathcal{I} \subseteq V \big\}$.

Corollary 1 simplifies the expression of $\mathcal{S}$ for specific graphs.

Corollary 1 *In the cases of the fused graph and the complete graph, we have*

$$\mathcal{S} = \Big\{ \boldsymbol{s} : \Big|\sum_{i=1}^{l} n_i s_i\Big| \le l\lambda_1 + \lambda_2, \quad \Big|\sum_{i=L-l+1}^{L} n_i s_i\Big| \le l\lambda_1 + \lambda_2, l = 1, \cdots, L-1,$$
$$\Big|\sum_{i=l_1+1}^{l_2} n_i s_i\Big| \le (l_2 - l_1)\lambda_1 + 2\lambda_2, 1 \le l_1 < l_2 < L; \quad \Big|\sum_{i=1}^{L} n_i s_i\Big| \le L\lambda_1 \Big\},$$
$$\mathcal{S} = \Big\{ \boldsymbol{s} : \Big|\sum_{i=1}^{l} n_{k_i} s_{k_i}\Big| \le l\lambda_1 + l(L-l)\lambda_2, \quad \Big|\sum_{i=L-l+1}^{L} n_{k_i} s_{k_i}\Big| \le l\lambda_1 + l(L-l)\lambda_2,$$
$$l = 1, \cdots, L, s_{k_1} \ge \cdots \ge s_{k_L} \Big\}.$$

The partition rule is useful for efficient computation, as it may reduce computation cost substantially. It can be used in several ways. First, the rule partitions nodes into disjoint connected subsets through the sample covariances s_{jkl}'s. This breaks the original large problem into smaller subproblems, owing to this necessary and sufficient rule. Second, **Algorithm 1** can be applied to each subproblem independently, permitting parallel computation.

Algorithm 2 integrates the partition rule in Theorem 3 with **Algorithm 1** to make the proposed method applicable to large-scale problems.

Algorithm 2 (A partition version of Algorithm 1):

Step 1. (Screening) Compute the sample-covariance matrix $\boldsymbol{S}_l$; $l = 1, \cdots, L$. Construct a $p \times p$ symmetric matrix $\boldsymbol{T} = (t_{jk})_{1 \le j,k \le p}$, with $t_{jk} = 0$ if $(s_{jk1}, \cdots, s_{jkL}) \in \mathcal{S}$ and $t_{jk} = 1$ otherwise. Treating $\boldsymbol{T}$ as an adjacency matrix of an undirected graph, we compute its maximum connected components to form a partition of nodes $\{\mathcal{J}_1, \cdots, \mathcal{J}_q\}$ using breadth-first search or depth-first search algorithm, c.f., [3]

Step 2. (Subproblems) For $i = 1, \cdots, q$, solve (2) for each subproblem consisting of nodes in $\mathcal{J}_i$, by applying **Algorithm 1** to obtain the solution $\hat{\boldsymbol{\Omega}}^{(i)} = \big(\hat{\boldsymbol{\Omega}}_1^{(i)}, \cdots, \hat{\boldsymbol{\Omega}}_L^{(i)}\big); i = 1, \cdots, q$.

Step 3. (Combining results) The final solution $\hat{\boldsymbol{\Omega}}_l = \mathrm{Diag}\big(\hat{\boldsymbol{\Omega}}_l^{(1)}, \cdots, \hat{\boldsymbol{\Omega}}_l^{(q)}\big)$; $l = 1, \cdots, L$.

4 Theory

This section investigates theoretical aspects of the proposed method. First we develop a general theory on maximum penalized likelihood estimation involving two types of L_0-constraints for pursuit of sparseness and clustering. Then we specialize the theory for estimation of multiple precision matrices in Section 4.3. Now consider a constrained L_0-version of (2):

$$\max_{\boldsymbol{\theta}=(\boldsymbol{\beta},\boldsymbol{\eta})} L(\boldsymbol{\theta}), \text{ subject to } \sum_{j=1}^{d} \mathbb{I}(|\beta_j| \neq 0) \leq C_1, \sum_{(jj')\in\mathcal{E}} \mathbb{I}\big(|\beta_j - \beta_{j'}| \neq 0\big) \leq C_2. \quad (13)$$

as well as its computational surrogate

$$\max_{\boldsymbol{\theta}=(\boldsymbol{\beta},\boldsymbol{\eta})} L(\boldsymbol{\theta}), \text{ subject to } \sum_{j=1}^{d} J_\tau(|\beta_j|) \leq C_1, \sum_{(jj')\in\mathcal{E}} J_\tau\big(\big|\beta_j - \beta_{j'}\big|\big) \leq C_2, \quad (14)$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta})$ with $\boldsymbol{\beta} \in \mathbb{R}^d$ and $\boldsymbol{\eta}$ representing the off-diagonals and diagonals of $\boldsymbol{\Omega}$, and three non-negative tuning parameters (C_1, C_2, τ). Note that **Algorithm 1** yields a local minimizer of (14), relaxing it by solving a sequence of convex problems.

In what follows, we will prove that global minimizers of (13) and (14) reconstruct the ideal *oracle estimator* as if the true sparseness and clustering structures of the precision matrices were known in advance. As a result of the reconstruction, key properties of the oracle estimator are simultaneously achieved by the proposed method.

4.1 The oracle estimator and consistent graph

To define the oracle estimator, let $\mathcal{G}(\boldsymbol{\beta})$ denote a partition of $\mathcal{I} \equiv \{1, \cdots, d\}$ by the parameter $\boldsymbol{\beta}$, i.e. $\mathcal{G}(\boldsymbol{\beta}) = (\mathcal{I}_0(\boldsymbol{\beta}), \cdots, \mathcal{I}_{K(\boldsymbol{\beta})}(\boldsymbol{\beta}))$, with $\mathcal{I}_0(\boldsymbol{\beta}) = \mathcal{I} \setminus A(\boldsymbol{\beta})$ and $\mathcal{I}_k(\boldsymbol{\beta})$ satisfying $\boldsymbol{\beta}_j = \boldsymbol{\beta}_{j'}$; $j, j' \in \mathcal{I}_k(\boldsymbol{\beta})$; $k = 1, \cdots, K(\boldsymbol{\beta})$, where $K(\boldsymbol{\beta})$ is the number of nonzero clusters and $A(\boldsymbol{\beta}) \equiv \{i : \beta_i \neq 0\}$ is the support of $\boldsymbol{\beta}$. Let $\mathcal{G}^0 = \mathcal{G}(\boldsymbol{\beta}^0)$ be the true partition induced by $\boldsymbol{\beta}^0$, with $\boldsymbol{\theta}^0 = (\boldsymbol{\beta}^0, \boldsymbol{\eta}^0)$ the true parameter value and $\boldsymbol{\beta}^0 \in \mathbb{R}^d$.

Definition 1 (Oracle estimator) *Given $\mathcal{G}^0$, the oracle estimator is defined as: $\hat{\boldsymbol{\theta}}^o = (\hat{\boldsymbol{\beta}}^o, \hat{\boldsymbol{\eta}}^o) = \mathrm{argmax}_{\boldsymbol{\beta}:\mathcal{G}(\boldsymbol{\beta})=\mathcal{G}^0} L(\boldsymbol{\theta})$, the corresponding maximum likelihood estimator.*

In (13) and (14), the edge set $\mathcal{E}$ of $\mathcal{U}$ is important for clustering. In order for simultaneous pursuit of sparseness and clustering structures to be possible, we may need $\mathcal{U}$ to be consistent with the clustering structure of the true precision matrices. In other words, a consistent graph is a minimal requirement for reconstruction of the oracle estimator, where there must exist a path connecting any nodes within the same true cluster.

Definition 2 (*Consistent* graph $\mathcal{U}$) *An undirected graph $\mathcal{U} = \big(\mathcal{I}, \mathcal{E}\big)$ is consistent with the true cluster $\mathcal{G}^0 = \{\mathcal{I}_0^0, \cdots, \mathcal{I}_{K_0}^0\}$, if the subgraph restricting nodes on $\mathcal{I}_j^0$ is connected; $j = 1, \cdots, K_0$.*

4.2 Non-asymptotic probability error bounds

This section derives a non-asymptotic probability error bound for simultaneous sparseness and clustering pursuit, based on which we prove that (13) and (14) reconstruct the oracle estimator. This implies consistent identification of the sparseness and clustering structures of multiple graphical models, under one simple assumption, called the degree-of-separation condition.

Before proceeding, we introduce some notations. Given a graph $\mathcal{U} = (\mathcal{I}, \mathcal{E})$, let $\mathcal{S} = \big\{\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta}) : |A(\boldsymbol{\beta})| \leq d_0, C(\boldsymbol{\beta}, \mathcal{E}) \leq c_0, \mathcal{G}(\boldsymbol{\beta}) \neq \mathcal{G}(\boldsymbol{\beta}^0)\big\}$ be a constrained set with $C(\boldsymbol{\beta}, \mathcal{E}) = \sum_{(jj')\in\mathcal{E}} \mathbb{I}\big(|\beta_j - \beta_{j'}| \neq 0\big)$, where $d_0 = |A^0|$ with $A^0 = A(\boldsymbol{\beta}^0)$ as defined above. Given a partition $\mathcal{G}$, let $\mathcal{S}_{\mathcal{G}} = \big\{\boldsymbol{\theta} \in \mathcal{S} : \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}\big\}$. Given an index set $A \subseteq \mathcal{I}$, let $\mathcal{S}_A = \big\{\boldsymbol{\theta} \in \mathcal{S} : \boldsymbol{A}(\boldsymbol{\beta}) = A\big\}$. Let $\mathcal{S}_i = \cup_{A:|A^0\setminus A|=i}\mathcal{S}_A$, $S_i^\star = \max_{A:|A^0\setminus A|=i} \big|\{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta}) \in \mathcal{S}_A\}\big|$; $i = 0, \cdots, d_0$, and $S^* = \exp\big(\max_{0\leq i\leq d_0} \frac{\log S_i^*}{\max(i,1)}\big)$. Roughly, $S^\star$ quantifies complexity of the space of candidate precision matrices scaled by the number of nonzero entries.

The degree-of-separation condition will be used to ensure consistent reconstruction of the oracle estimator: For some constant $c_1 > 0$,

$$C_{\min}(\boldsymbol{\theta}^0) \geq c_1 \frac{\log d + \log S^*}{n}, \tag{15}$$

where $C_{\min}(\boldsymbol{\theta}^0) \equiv \inf_{\{\boldsymbol{\theta}=(\boldsymbol{\beta},\boldsymbol{\eta})\in\mathcal{S}\}} \frac{-\log(1-h^2(\boldsymbol{\theta},\boldsymbol{\theta}^0))}{\max(|A^0\setminus A(\boldsymbol{\beta})|,1)}$ with $|\cdot|$ and $\setminus$ denoting the size of a set and that of set difference, respectively, $h(\boldsymbol{\theta},\boldsymbol{\theta}^0) = \big(\frac{1}{2}\int(g^{1/2}(\boldsymbol{\theta},y) - g^{1/2}(\boldsymbol{\theta}^0,y))^2 d\mu(y)\big)^{1/2}$ is the Hellinger-distance for densities with respect to a dominating measure μ.

We now define the bracketing Hellinger metric entropy of space $\mathcal{F}$, denoted by the function $H(\cdot,\mathcal{F})$, which is the logarithm of the cardinality of the u-bracketing (of $\mathcal{F}$) of the smallest size. That is, for a bracket covering $S(\varepsilon,m) = \{f_1^l, f_1^u, \cdots, f_m^l, f_m^u\} \subset \mathcal{L}_2$ satisfying $\max_{1\le j\le m}\|f_j^u - f_j^l\|_2 \le \varepsilon$ and for any $f\in\mathcal{F}$, there exists a j such that $f_j^l \le f \le f_j^u$, *a.e.* P, then $H(u,\mathcal{F})$ is $\log(min\{m:\ S(u,m)\})$, where $\|f\|_2 = \int f^2(z)d\mu$. For more discussions about metric entropy of this type, see [8].

Assumption A: (Complexity of the parameter space) For some constant $c_0 > 0$ and any $0 < t < \varepsilon \le 1$, $H(t,\mathcal{B}_{\mathcal{G}}) \le c_0(\log p)^2, 1)|A|\log(2\varepsilon/t)$, where $\mathcal{B}_{\mathcal{G}} = \mathcal{F}_{\mathcal{G}} \cap \{h(\boldsymbol{\theta},\boldsymbol{\theta}^0) \le 2\varepsilon\}$ is a local parameter space, and $\mathcal{F}_{\mathcal{G}} = \{g^{1/2}(\boldsymbol{\theta},y) : \boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{\eta}) : \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}\}$ be a collection of square-root densities indexed by any subset $\mathcal{G} \in \{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{\eta}) \in \mathcal{S}\}$.

Next we present our non-asymptotic probability error bounds for reconstruction of the oracle estimator $\hat{\boldsymbol{\theta}}^0$ by global minimizers of (13) and (14) in terms of $C_{min}(\boldsymbol{\theta}^0)$, n, d and d_0, where d_0 and d can depend on n. Consistency is established for reconstruction of $\hat{\boldsymbol{\theta}}^0$ as well as structure recovery. Note that $\hat{\boldsymbol{\theta}}^0$ is asymptotically optimal, hence the optimality translates into the global minimizers of (13) and (14).

Theorem 4 *(Global minimizer of* (13)*) Under* **Assumption A**, *if* $\mathcal{U}$ *is consistent with* $\mathcal{G}^0$, *then for a global minimizer of (13)* $\hat{\boldsymbol{\theta}}^{l_0}$ *with estimated grouping* $\hat{\mathcal{G}}^{l_0} = \mathcal{G}(\hat{\boldsymbol{\beta}}^{l_0})$ *at* $(C_1,C_2) = (d_0,c_0)$ *with* $c_0 = C(\boldsymbol{\beta}^0,\mathcal{E})$,

$$\mathbb{P}\Big(\hat{\mathcal{G}}^{l_0} \ne \mathcal{G}^0\Big) = \mathbb{P}\Big(\hat{\boldsymbol{\theta}}^{l_0} \ne \hat{\boldsymbol{\theta}}^{o}\Big) \ \le\ \exp\Big(-c_2 n C_{\min}(\boldsymbol{\theta}^0) + 2\log d + \log S^{\star}\Big). \tag{16}$$

Under (15), $\mathbb{P}\Big(\hat{\mathcal{G}}^{l_0} = \mathcal{G}^0\Big) = \mathbb{P}\Big(\hat{\boldsymbol{\theta}}^{l_0} = \hat{\boldsymbol{\theta}}^{o}\Big) \to 1$ *as* $n, d \to \infty$.

For the constrained truncated L_1-likelihood, one additional condition–**Assumption B** is necessary. We requires the Hellinger-distance to be smooth so that the approximation of

the truncated L_1-function to the L_0-function becomes adequate by tuning τ.

Assumption B: For some constants d_1-$d_3 > 0$,

$$-\log(1-h^2(\boldsymbol{\theta},\boldsymbol{\theta}^0)) \geq -d_1\log(1-h^2(\boldsymbol{\theta}^\tau,\boldsymbol{\theta}^0)) - d_3\tau^{d_2}d, \tag{17}$$

where $\boldsymbol{\theta}^\tau = (\boldsymbol{\beta}^\tau, \eta)$ with $\boldsymbol{\beta}^\tau = (\beta_1^\tau, \cdots, \beta_p^\tau)$, and $\beta_j^\tau = \frac{\sum_{j'\in\mathcal{I}_k}\beta_j'}{|\mathcal{I}_k|}$ for $j \in \mathcal{I}_k(\boldsymbol{\beta})$; $k = 0, 1, \cdots, K(\boldsymbol{\beta})$.

Theorem 5 *(Global minimizer of* (14)*) Assume that* **Assumption A** *with* $\mathcal{F}_\mathcal{G}$ *replaced by* $\mathcal{F}_\mathcal{G}^\tau = \{g^{1/2}(\boldsymbol{\theta}, y) : \boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta}) : \mathcal{G}^\tau(\boldsymbol{\beta}) = \mathcal{G}\}$ *and* **Assumption B** *are met. If* $\mathcal{U}$ *is consistent with* $\mathcal{G}^0$*, then for a global minimizer of* (14) $\hat{\boldsymbol{\theta}}^g$ *with estimated grouping* $\hat{\mathcal{G}}^g = \mathcal{G}(\hat{\boldsymbol{\beta}}^g)$ *at* $(C_1, C_2) = (d_0, c_0)$ *with* $c_0 = C(\boldsymbol{\beta}^0, \mathcal{E})$ *and* $\tau \leq \big(\frac{(d_1-c_3)C_{\min}(\boldsymbol{\theta}^0)}{d_3 d}\big)^{1/d_2}$,

$$\mathbb{P}\Big(\hat{\mathcal{G}}^g \neq \mathcal{G}^0\Big) = \mathbb{P}\Big(\hat{\boldsymbol{\theta}}^g \neq \hat{\boldsymbol{\theta}}^o\Big) \;\leq\; \exp\Big(-c_3 n C_{\min}(\boldsymbol{\theta}^0) + 2\log d + \log S^\star\Big). \tag{18}$$

Under (15), $\mathbb{P}\Big(\hat{\mathcal{G}}^g = \mathcal{G}^0\Big) = \mathbb{P}\Big(\hat{\boldsymbol{\theta}}^g = \hat{\boldsymbol{\theta}}^o\Big) \to 1$ *as* $n, d \to \infty$.

4.3 An illustrative example

We now apply the general theory in Theorems 2 and 3 to the estimation of multiple precision matrices, in which the true precision matrices in each cluster are the same, with $g_0 \equiv \sum_{l=1}^{L-1}\sum_{j>k}\mathbb{I}\big(\omega_{jkl}^0 \neq \omega_{jk(l+1)}^0\big)$ the number of break points among these clusters. In this case, a serial graph $\mathcal{U}$ is considered for clustering.

Denote by p and L_0 the dimension of the precision matrix and the number of distinctive clusters, respectively. Let $H_l = \Big(\frac{\partial^2(-\log\det(\boldsymbol{\Omega}_l))}{\partial^2\boldsymbol{\Omega}}\Big)\Big|_{\boldsymbol{\Omega}_l=\boldsymbol{\Omega}_l^0}$ be the $p^2 \times p^2$ Hessian matrix of $-\log\det(\boldsymbol{\Omega}_l)$, whose $(jk, j'k')$ element is $tr(\boldsymbol{\Sigma}_l\Delta_{jk}\boldsymbol{\Sigma}_l\Delta_{j'k'})$, c.f., [2]. Define

$$\eta_{min} = \min\Big(\min_{(j,k,l):\omega_{jkl}^0\neq 0}\big|\omega_{jkl}^0\big|, \frac{1}{\sqrt{2}}\min_{(j,k,l):\omega_{jkl}^0\neq\omega_{jk(l+1)}^0}\big|\omega_{jkl}^0 - \omega_{jk(l+1)}^0\big|\Big)$$

to be the resolution level for simultaneous sparseness and clustering pursuit.

An application of Theorems 2 and 3 with $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ being off-diagonals and diagonals of $\boldsymbol{\Omega}$ leads to the following result.

Corollary 2 *(Multiple precision matrices with a serial graph) When* $\mathcal{U}$ *is a serial graph, all*

the results in Theorems 2 and 3 for simultaneous pursuit of sparseness and clustering hold under two simple conditions:

$$C_{min}(\boldsymbol{\theta}^0) \geq c_4 \min_{1\leq l\leq L} c_{min}\big(H_l\big)\eta_{min}^2, \ \textit{and} \ \log S^\star \leq 2g_0 \max(\log(d_0/g_0), 1), \tag{19}$$

for some constant $c_4 > 0$. *Sufficiently, if*

$$\min_{1\leq l\leq L} c_{min}\big(H_l\big)\eta_{min}^2 \geq c_0 \frac{\log\big(Lp(p-1)/2\big) - g_0 \max\big(\log(d_0/g_0), 1\big)}{n}, \tag{20}$$

holds for some constant $c_0 > 0$, *then* $\mathbb{P}\Big(\hat{\boldsymbol{\Omega}}^{\ell_0} \neq \hat{\boldsymbol{\Omega}}^o\Big)$ *and* $\mathbb{P}\Big(\hat{\boldsymbol{\Omega}}^{g} \neq \hat{\boldsymbol{\Omega}}^o\Big) \to 0$ *as* $n, d \to +\infty$.

Corollary 2 suggests that the amount of reconstruction improvement would be of the order of $1/L$ if the L precision matrices are identical. In general, the amount of improvement of joint estimation over separate estimation is $L/\log(L)$ when g_0 is small, i.e. $g_0 \max\big(\log(d_0/g_0), 1\big) \lesssim \log\big(Lp(p-1)/2\big)$, by contrasting the sufficient condition in (20) with that for a separate estimation approach in [17], where $\lesssim$ denotes inequality ignoring constant terms. Here g_0 describes similarity among L precision matrices with a small value corresponding to a high-degree of similarity shared among precision matrices.

5 Numerical examples

This section studies operational characteristics of the proposed method via simulation in sparse and nonsparse situations with different types of graphs in both low- and high-dimensional settings. In each simulated example, we compare our method against its convex counterpart for seeking the sparseness structure for each graphical model and identifying the grouping structure among multiple graphical models, and contrast the method against its counterpart seeking the sparseness structure alone. In addition, we also compare against a kernel smoothing method for time-varying networks [7, 25] in Examples 1-3, whenever appropriate. The smoothing method defines a weighted average over sample covariance matrices at time points as $\tilde{\boldsymbol{S}}_l(h) = \frac{\sum_{l'=1}^L w_{ll'}(h)\boldsymbol{S}_l'}{\sum_{l'=1}^L w_{ll'}(h)}$, with $w_{ll'}(h) = K(h^{-1}|l-l'|)$; $l = 1, \cdots, L$, where $K(x) = (1-|x|)\mathbb{I}(|x| < 1)$ is a triangular kernel, h is a bandwidth, and $l = 1, \cdots, L$ denotes

clusters. Then within each cluster l, the precision matrix estimate $\hat{\mathbf{\Omega}}_l(h,\lambda)$ is obtained by solving

$$\hat{\mathbf{\Omega}}_l(h,\lambda) \;=\; \underset{\mathbf{\Omega}_l \succ 0}{\operatorname{argmin}} \Big(-\log\det\big(\mathbf{\Omega}_l\big) + tr(\mathbf{\Omega}\tilde{\boldsymbol{S}}_l(h)) + \lambda \sum_{j<j'} |\omega_{jj'l}| \Big), l = 1, \cdots, L \quad (21)$$

using the glasso algorithm [5], and the final estimate is obtained through tuning over (h,λ)-grids. Two performance metrics are used to measure the accuracy of parameter estimation as well as that of correct identification of the sparseness and grouping structures.

In Examples 1-3, temporal clustering pursuit is performed over $\mathbf{\Omega}_1, \cdots, \mathbf{\Omega}_L$ through a serial graph $\mathcal{E} = \big\{\{(j,k,l),(j',k',l')\} : j = j', k = k', |l - l'| = 1\big\}$. That is, only adjacent matrices may be possibly clustered. In Example 4, general clustering pursuit is conducted through a complete graph $\mathcal{E} = \big\{\{(j,k,l),(j',k',l')\} : j = j', k = k', l < l'\big\}$.

For the accuracy of parameter estimation, the average entropy loss (EL) and average quadratic loss (QL) are considered, defined as

$$EL = \frac{1}{L}\sum_{l=1}^{L}\Big(tr\big(\mathbf{\Omega}_l^{-1}\hat{\mathbf{\Omega}}_l\big) - \log\det\big(\mathbf{\Omega}_l^{-1}\hat{\mathbf{\Omega}}_l\big)\Big), \quad QL = \frac{1}{L}\sum_{l=1}^{L} tr\Big(\big(\mathbf{\Omega}_l^{-1}\hat{\mathbf{\Omega}}_l - \boldsymbol{I}\big)^2\Big).$$

For the accuracy of identification, average false positive (FPV) and false negative (FNV) rates for sparseness pursuit, as well as those (FPG) and (FNG) for grouping are used:

$$\begin{aligned}
FPV &= \frac{1}{L}\sum_{l=1}^{L} \frac{\sum_{1\le j<j'\le p}\mathbb{I}(\omega_{jj'l}=0, \hat{\omega}_{jj'l}\neq 0)}{\sum_{1\le j<j'\le p}\mathbb{I}(\omega_{jj'l}=0)}\Big(1 - \mathbb{I}\big(\mathbf{\Omega}_{l,\text{off}} \neq \mathbf{0}\big)\Big) \\
FNV &= \frac{1}{L}\sum_{l=1}^{L} \frac{\sum_{1\le j<j'\le p}\mathbb{I}(\omega_{jj'l}\neq 0, \hat{\omega}_{jj'l}=0)}{\sum_{1\le j<j'\le p}\mathbb{I}(\omega_{jj'l}\neq 0)}\mathbb{I}\big(\mathbf{\Omega}_{l,\text{off}} \neq \mathbf{0}\big), \\
FPG &= \frac{1}{|\mathcal{E}|}\sum_{l\sim l'} \frac{\sum_{1\le j<j'\le p}\mathbb{I}(\omega_{jj'l}=\omega_{jj'l'}, \hat{\omega}_{jj'l}\neq\hat{\omega}_{jj'l'})}{\sum_{1\le j<j'\le p}\mathbb{I}(\omega_{jj'l}=\omega_{jj'l'})}\Big(1 - \mathbb{I}\big(\mathbf{\Omega}_{l,\text{off}} \neq \mathbf{\Omega}_{l',\text{off}}\big)\Big), \\
FNG &= \frac{1}{|\mathcal{E}|}\sum_{l\sim l'} \frac{\sum_{1\le j<j'\le p}\mathbb{I}(\omega_{jj'l}\neq\omega_{jj'l'}, \hat{\omega}_{jj'l}=\hat{\omega}_{jj'l'})}{\sum_{1\le j<j'\le p}\mathbb{I}(\omega_{jj'l}\neq\omega_{jj'l'})}\mathbb{I}\big(\mathbf{\Omega}_{l,\text{off}} \neq \mathbf{\Omega}_{l',\text{off}}\big),
\end{aligned}$$

where $\mathbf{\Omega}_{l,\text{off}}$ denotes the off-diagonal elements of $\mathbf{\Omega}_l$. Note that FPV and FNG as well as FNV and FNG are not comparable due to normalization with and without the zero-group, respectively.

For tuning, we minimize a prediction criterion with respect to the tuning parameter(s) on an independent test set with the same sample size as the training set. The prediction criterion is $CV(\boldsymbol{\lambda}) = \frac{1}{L}\sum_{l=1}^{L}\Big(-\log\det\big(\hat{\boldsymbol{\Omega}}_l(\boldsymbol{\lambda})\big) + tr\big(\boldsymbol{S}_l^{\text{tune}}\hat{\boldsymbol{\Omega}}_l(\boldsymbol{\lambda})\big)\Big)$, where $\boldsymbol{S}_l^{\text{tune}}$ is the sample covariance matrix for the tuning data; $l = 1, \cdots, L$. Then the estimated tuning parameter is obtained: $\boldsymbol{\lambda}^{\star} = \text{argmin}_{\boldsymbol{\lambda}}\, CV(\boldsymbol{\lambda})$, which is used in the estimated precision matrices. Here minimization of $CV(\boldsymbol{\lambda})$ is performed through a simple grid search over the domain of the tuning parameter(s).

All simulations are performed based on 100 simulation replications. Three different types of networks are considered. Specifically, Example 1 concerns a chain network with small p and L but large n, where each $\boldsymbol{\Omega_l}$ is relatively sparse and a temporal change occurs at two different l values. Example 2 deals with a nearest neighbor networks for each Ω_l and the same temporal structure as in Example 2. Examples 3 and 4 study exponentially decaying networks in nonsparse precision matrices in high and low-dimensional situations with large and small L, respectively. In Examples 1-3 and Example 4, **Algorithms** 1 and 2 are respectively applied.

Example 1: Chain networks: This example estimates tridiagonal precision matrices as in [4]. Specifically, $\boldsymbol{\Omega}_l^{-1} = \boldsymbol{\Sigma}_l$ is AR(1)-structured with its ij-element being $\sigma_{ijl} = \exp(-|s_{il} - s_{jl}|/2)$, and $s_{1l} < s_{2l} < \cdots < s_{pl}$ are randomly chosen: $s_{il} - s_{(i-1)l} \sim \text{Unif}(0.5,\, 1)$; $i = 2, \cdots, p$, $l = 1, \cdots, L$. The following situations are considered: (I) $(n, p, L) = (120, 30, 4)$, $(n, p, L) = (120, 200, 4)$, with $\boldsymbol{\Omega}_1 = \boldsymbol{\Omega}_2$, $\boldsymbol{\Omega}_3 = \boldsymbol{\Omega}_4$; (II) $(n, p, L) = (120, 20, 30)$, $(n, p, L) = (120, 10, 90)$, with $\boldsymbol{\Omega}_1 = \cdots = \boldsymbol{\Omega}_{L/3}$, $\boldsymbol{\Omega}^{(1+L/3)} = \cdots = \boldsymbol{\Omega}_{2L/3}$, $\boldsymbol{\Omega}_{1+2L/3} = \cdots = \boldsymbol{\Omega}_L$. Then, we study the proposed method's performance as a function of the number of graphs and the number of nodes.

Example 2: Nearest neighbor networks. This example concerns networks described in [11]. In particular, we generate p points randomly on a unit square, and compute the k nearest neighbors of each point based on the Euclidean distance. In the case of $k = 3$, three

points are connected to each point. For each "edge" in the graph, the corresponding off-diagonal in a precision matrix is sampled independently according to the uniform distribution over $[-1,-0.5]\cup[0.5,1]$, and the ith diagonal is set to be the sum of the absolute values of the ith row off-diagonals. Given the previous cluster, the matrices in the current cluster are obtained by randomly adding or deleting a small fraction of nonzero elements in the matrices from previous cluster. Finally, each row of a precision matrix is divided by the square root of the product of corresponding diagonals ($\omega_{ij} \leftarrow \frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}$) so that diagonals of the final precision matrices are one. The following scenarios are considered: (I) $(n,p,L)=(300,30,4)$ and $(n,p,L)=(300,200,4)$, where $\mathbf{\Omega}_1=\mathbf{\Omega}_2$, $\mathbf{\Omega}_3=\mathbf{\Omega}_4$, and (II) $(n,p,L)=(300,20,30)$ and $(n,p,L)=(300,10,90)$, where $\mathbf{\Omega}_1=\cdots=\mathbf{\Omega}_{L/3}, \mathbf{\Omega}_{1+L/3}=\cdots=\mathbf{\Omega}_{2L/3}, \mathbf{\Omega}_{1+2L/3}=\cdots=\mathbf{\Omega}_l$. In (I), the first cluster of matrices $(\mathbf{\Omega}_1,\mathbf{\Omega}_2)$ are generated using the above mechanism, with the second cluster of matrices $(\mathbf{\Omega}_3,\mathbf{\Omega}_4)$ obtained by deleting one edge for each node in the network. In (II), the generating mechanism remains except that the third cluster of matrices $(\mathbf{\Omega}_{1+2L/3},\cdots,\mathbf{\Omega}_L)$ are generated by adding an edge for each node in its previous adjacent network.

Example 3: Exponentially decaying networks. This example examines a nonsparse situation in which elements of precision matrices are nonzero, and decay exponentially with respect to their Euclidean distances to the corresponding diagonals. In particular, the (i,j)th entry of the lth precision matrix ω_{ijl} is $\exp\big(a_l|i-j|\big)$ with a_l sampled uniformly over $[1,2]$. In this case, it is sensible to report the results for parameter estimation as opposed to identifying nonzeros As in Examples 1 and 2, several scenarios are considered: (I) $(p,L)=(30,4)$, $(p,L)=(200,4)$, and the sample size $n=120$ or 300 with $\mathbf{\Omega}_1=\mathbf{\Omega}_2$, $\mathbf{\Omega}_3=\mathbf{\Omega}_4$, and (II) $(p,L)=(20,30)$, $(p,L)=(10,90)$, and the sample size $n=120$ or 300 with $\mathbf{\Omega}_1=\cdots=\mathbf{\Omega}_{L/3}, \mathbf{\Omega}_{1+L/3}=\cdots=\mathbf{\Omega}_{2L/3}, \mathbf{\Omega}_{1+2L/3}=\cdots=\mathbf{\Omega}_L$.

Example 4: Large precision matrices. This example utilizes the partition rule to treat large-scale simulations. First, we examine two cases $(n,p,L)=(120,1000,4)$ and

$(n, p, L) = (500, 2000, 4)$ with $\mathbf{\Omega}_1 = \mathbf{\Omega}_2$ and $\mathbf{\Omega}_3 = \mathbf{\Omega}_4$, where four precision matrices are considered with size 1000×1000 and 2000×2000 for pairwise clustering, where $\mathcal{U}$ is the complete graph. Here each precision matrix is set to be a block-diagonal matrix: $\mathbf{\Omega}_l = \text{Diag}(\mathbf{\Omega}_{l1}, \cdots, \mathbf{\Omega}_{lq})$; $l = 1, \cdots, L$, where $\mathbf{\Omega}_{1j} = \mathbf{\Omega}_{2j}, \mathbf{\Omega}_{3j} = \mathbf{\Omega}_{4j}$ are 20×20 matrices generated in the same fashion as that in **Examples 1**. Finally, the complete graph is used as opposed to the fused graph. Overall, the complexity is much higher than the previous examples.

Tables 1-4 and Figure 1 about here

As suggested by Tables 1-4, the proposed method performs well against its competitors in parameter estimation and correct identification of the sparseness and grouping structures across all the situations. With regard to accuracy of identification of the sparseness and clustering structures, the proposed method has the smallest false positives in terms of FPV and FPG, yielding sharper parameter estimation than the competitors. This says that shrinkage towards common elements is advantageous for parameter estimation in a low-or high-dimensional situation. Note that the largest improvement occurs for the most difficult situation in Example 4.

Compared with pursuit of sparseness alone–TLP, the amount of of improvement of our method is from 143% to 244% and 118% % to 236% in terms of the EL and QL when $n = 120$, and from 80.5% to 228% and 96.5% to 240% in terms of the EL and QL when $n = 300$, as indicated in Table 3. This comparison suggests that exploring the sparseness structure alone is inadequate for multiple graphical models. Pursuit of two types of structures appears advantageous in terms of performance, especially for large matrices.

Compared with its convex counterpart "our-con", our method leads to between a 19.9% and a 106% improvement, and between a 4.2% improvement and a 75.5% improvement in terms of the EL and QL when $n = 120$, and between a 18.3% improvement and a 120% improvement, and a 90.3% improvement and a 151% improvement in terms of the EL and

QL when $n = 300$; see Table 3. This is expected because more accurate identification of structures tends to yield better parameter estimation.

In contrast to the smoothing method [7, 25] for time-varying network analysis, across all cases except one low-dimensional case of $L = 4$ and $p = 30$ in Table 3, our method yields a 54.5% improvement and a 20.3% improvement in terms of the EL and QL when $n = 120$, and a 51.9% improvement and a 25.8% improvement in terms of the EL and QL when $n = 300$ when L is not too small, c.f. Tables 1-4.

To understand how the proposed method performs relative to (n, p, L), we examine Table 3 and Figure 1 in further detail. Overall, the proposed method performs better as n, L increases and worse as p increases. Interestingly, as suggested by Figure 1, the method performs better as L increases, which confirms with our theoretical analysis.

In summary, the proposed method achieves the desired objective of pursuing simultaneous both sparseness and clustering structures to battle the curse of dimensionality in a high-dimensional situation.

6 Signaling network inference

This section applies the proposed method to the multivariate single cell flow cytometry data in [16] to infer a signaling network or pathway; a consensus version of the network with eleven proteins is described in Figure 3. In this study, a multiparameter flow cytometry recorded the quantitative amounts of the eleven proteins in a single cell as an observation. To infer the network, experimental perturbations on various aspects of the network were imposed before the amounts of the eleven proteins were measured under each condition. The idea was that, if a chemical was applied to stimulate or inhibit the activity of a protein, then both the abundance of the protein and those of its downstream proteins in the network would be expected to increase or decrease, while those of non-related proteins would barely change. There were ten types of experimental perturbations on different targets: 1) activating a

target (CD3) in the upstream of the network so that the whole network was expected to be perturbed; 2) activating a target (CD28) in the upsteam of the network; 3) activating a target (ICAM2) in the upsteam of the network; 4) activating PKC; 5) activating PKA; 6) inhibiting PKC; 7) inhibiting Akt; 8) inhibiting PIP2; 9) inhibiting Mek; 10) inhibiting a target (PI3K) in the upstream of the network. In [16], data were collected under nine experimental conditions and then used to infer a directed network; each of the nine experimental conditions was either a single type of perturbation or a combination of two or three types of perturbations. Interestingly, data were also collected under another five conditions, each of which was a combination of two of the previous nine conditions. Hence, the data offered an opportunity to infer the two networks under the two sets of the conditions: since the two sets of conditions largely overlapped, we would expect the two networks to be largely similar to each other; on the other hand, due to the difference between the two sets of the conditions, some deviations between the two networks were also anticipate. There were $n_1 = 7466$ and $n_2 = 4206$ observations under the two sets of the conditions respectively.

We apply the proposed method to the normalized data under the two sets of the conditions respectively. Due to the expected similarities between the two networks, we consider grouping to encourage common structure defined by connecting edges between the two networks. The tuning parameters are estimated by a three-fold cross-validation. The reconstructed two undirected networks are now displayed in Figure 2, with 9 and 8 estimated (undirected) links for the two groups of conditions, being a subset of the 20 (directed) links in the gold standard signaling network as displayed in Figure 3, which is a consensus network that has been verified biologically, c.f. [16]. The reconstructed undirected graphs miss some edges as compared to the gold standard network, for instance, the links from protein "PKC" to "Raf" and "Mek". The three edges missed by [16], "PIP3" to "Akt", "Plcg" to "PKC", and "PIP2" to "PKC", are also missed by our method, possibly reflecting lack of information in the data due to no direct interventions imposed on "PIP3" and "Plcg".

Overall, the proposed method appears to work well in that the network inferred from the first set of conditions recovers one more dependence relationships than that from the second set of conditions, which is expected given that the second set of interventional conditions is less specific than the first one.

Here we analyze the data by contrasting the network constructed under the nine conditions with $n_1 = 7466$ against that under the five conditions with $n_2 = 4206$. Of particular interest is the detection of network structural changes between the two sets of conditions.

Figures 2 and 3 about here

7 Discussion

This article proposes a novel method to pursue two disparate types of structures—sparseness and clustering for multiple Gaussian graphical models. The proposed method is equipped with an efficient algorithm for large graphs, which is integrated with a partition rule to break down a large problem into many separate small problems to solve. For data analysis, we have considered signaling network inference in a low-dimensional situation. Worthy of note is that the proposed method can be equally applied to high-dimensional data, such as reconstructing and comparing gene regulatory networks across four subtypes of glioblastoma multiforme based on gene expression data [21].

To make the proposed method useful in practice, inferential tools need to be further developed. A Monte Carlo method may be considered given the level of complexity of the underlying problems. Moreover, the general approach developed here can be expanded to other types of graphical models, for instance, dynamic network models or time-varying graphical models [7]. This enables us to build time dependency into a model through, for example, a Markov property. Further investigation is necessary.

8 Appendix

Proof of Theorem 1: The equivalence follows directly from Theorem 4.1 in [20]. □

Next we present two lemmas to be used in the proof of Theorem 2.

Lemma 1 *For any* $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_m \in \mathbb{R}^L$, (22) *and* (23) *are equivalent:*

$$\exists\ |b_1|, \cdots, |b_m| \leq 1 \ s.t.\ \boldsymbol{x}_0 + b_1\boldsymbol{x}_1 + \cdots + b_m\boldsymbol{x}_m = \boldsymbol{0}, \tag{22}$$

$$for\ \forall\ \boldsymbol{c} \in \mathbb{R}^L, |\boldsymbol{c}^T\boldsymbol{x}_0| \leq |\boldsymbol{c}^T\boldsymbol{x}_1| + \cdots + |\boldsymbol{c}^T\boldsymbol{x}_m|. \tag{23}$$

Proof: If (22) holds, then for any $\boldsymbol{c} \in \mathbb{R}^L$, $|\boldsymbol{c}^T\boldsymbol{x}_0| = |b_1\boldsymbol{c}^T\boldsymbol{x}_1 + \cdots + b_m\boldsymbol{c}^T\boldsymbol{x}_m| \leq |b_1||\boldsymbol{c}^T\boldsymbol{x}_1| + \cdots + |b_m||\boldsymbol{c}^T\boldsymbol{x}_m|$, which is no greater than $|\boldsymbol{c}^T\boldsymbol{x}_1| + \cdots + |\boldsymbol{c}^T\boldsymbol{x}_m|$, implying (23). For the converse, assume that for any $\boldsymbol{c} \in \mathbb{R}^L$, $|\boldsymbol{c}^T\boldsymbol{x}_0| \leq |\boldsymbol{c}^T\boldsymbol{x}_1| + \cdots + |\boldsymbol{c}^T\boldsymbol{x}_m|$. Consider the following convex minimization:

$$\min_{\{b_1,\cdots,b_m\}} \sum_{i=1}^{m} B(b_i) \quad \text{subject to } \boldsymbol{x}_0 + b_1\boldsymbol{x}_1 + \cdots + b_m\boldsymbol{x}_m = \boldsymbol{0}, \tag{24}$$

where $B(x)$ is an indicator function with $B(x) = 0$ when $|x| \leq 1$ and $B(x) = +\infty$ otherwise. First, we need to show that the constraint set in (24) is nonempty. Suppose that it is empty. Let $\boldsymbol{c}_0 = (\boldsymbol{I} - \boldsymbol{P}_{(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m)})\boldsymbol{x}_0$, where $\boldsymbol{P}_{(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m)}$ is the projection matrix onto the linear space spanned by $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$. Since $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$ are linearly independent, we have that $\|\boldsymbol{c}_0\|_2 > 0$. Therefore $|\boldsymbol{c}_0^T\boldsymbol{x}_0| = \|\boldsymbol{c}_0\|_2^2 > 0 = |\boldsymbol{c}_0^T\boldsymbol{x}_1| + \cdots + |\boldsymbol{c}_0^T\boldsymbol{x}_m|$, contracting to that $|\boldsymbol{c}^T\boldsymbol{x}_0| \leq |\boldsymbol{c}^T\boldsymbol{x}_1| + \cdots + |\boldsymbol{c}^T\boldsymbol{x}_m|$. Hence the constraint set of (24) is nonempty and we denote its optimal value by $p^\star$. Next we convert (24) to its dual by introducing dual variable $\boldsymbol{\nu} \in \mathbb{R}^L$ for the equality constraints in (24) through Lagrange multipliers:

$$\max_{\{\boldsymbol{\nu}\in\mathbb{R}^L\}} \boldsymbol{\nu}^T\boldsymbol{x}_0 - |\boldsymbol{\nu}^T\boldsymbol{x}_1| - \cdots - |\boldsymbol{\nu}^T\boldsymbol{x}_m|. \tag{25}$$

By the assumption that $|\boldsymbol{c}^T\boldsymbol{x}_0| \leq |\boldsymbol{c}^T\boldsymbol{x}_1| + \cdots + |\boldsymbol{c}^T\boldsymbol{x}_m|$ for any $\boldsymbol{c}$, the maximal of (25) $d^\star$ must satisfy $d^\star \leq 0$. Hence $d^\star = 0$ because it is attained by $\boldsymbol{\nu} = \boldsymbol{0}$. Moreover, Slater's condition holds because constraint set of (24) is nonempty. By the strong duality principle, the duality gap is zero, and hence that $p^\star = d^\star = 0$. Consequently, a minimizer of (24) $(b_1, \cdots, b_m)$ exists with $|b_1| \leq 1, \cdots, |b_m| \leq 1$, satisfying the constraints $\boldsymbol{x}_0 + b_1\boldsymbol{x}_1 + \cdots + b_m\boldsymbol{x}_m = \boldsymbol{0}$. This implies (23). This completes the proof. □

Lemma 2 *For $\boldsymbol{s} = (s_1, \cdots, s_L)$ and a connected graph $\mathcal{G} = (V, \mathcal{E})$, there exist $|g_l| \leq 1$, $|g_{ll'}| \leq 1$, $g_{ll'} = -g_{l'l}$; $1 \leq l, l' \leq L$ such that*

$$\begin{cases} n_1 s_1 + \lambda_1 g_1 + \lambda_2 \sum_{l' \sim 1} g_{1l'} & = & 0 \\ \vdots & & \vdots \\ n_L s_L + \lambda_1 g_L + \lambda_2 \sum_{l' \sim L} g_{Ll'} & = & 0, \end{cases} \tag{26}$$

is equivalent to $|\sum_{l \in \mathcal{I}} n_l s_l| \leq \lambda_1 |\mathcal{I}| + \lambda_2 d(\mathcal{I}, \mathcal{I}^c)$ for any $\mathcal{I} \subseteq V$ with $d(\mathcal{I}, \mathcal{I}^c) = \sum_{l \in \mathcal{I}, l' \in \mathcal{I}^c} \mathbb{I}\big(l \sim l'\big)$.

Proof: First, for some $|g_l| \leq 1$, $|g_{ll'}| \leq 1, g_{ll'} = -g_{l'l}$; $1 \leq l, l' \leq L$, if (26) holds then,

$$|\sum_{l \in \mathcal{I}} n_l s_l| \;=\; \lambda_1 \Big| \sum_{l=1} g_l \Big| + \lambda_2 \Big| \sum_{l \in \mathcal{I}} \sum_{l' \sim l} g_{ll'} \Big| = \lambda_1 \Big| \sum_{l=1} g_l \Big| + \lambda_2 \Big| \sum_{l \in \mathcal{I}} \sum_{l' \in \mathcal{I}^c} \mathbb{I}(l \sim l') g_{ll'} \Big|,$$

which is no greater than $\lambda_1 |\mathcal{I}| + \lambda_2 d(\mathcal{I}, \mathcal{I}^c)$ for any $\mathcal{I} \subseteq V$. Conversely, by Lemma 1, it suffices to show that for any $\boldsymbol{c} \in \mathbb{R}^L$,

$$|\sum_{l=1}^{L} c_l n_l s_l| \leq \lambda_1 \sum_{l=1}^{L} |c_l| + \lambda_2 \sum_{l \sim l'} |c_l - c_{l'}| \tag{27}$$

provided that $|\sum_{l \in \mathcal{I}} n_l s_l| \leq \lambda_1 |\mathcal{I}| + \lambda_2 d(\mathcal{I}, \mathcal{I}^c)$ for any $\mathcal{I} \subseteq V$. To this end, for any permutation $(k_1, \cdots, k_L) \in \sigma(1, \cdots, L)$ and $l = 1, \cdots, L$, define convex region $\mathcal{C}_{lk_1 \cdots k_L} = \{\boldsymbol{c} = (c_1, \cdots, c_L) : c_{k_1} \geq \cdots \geq c_{k_l} \geq 0 \geq \cdots \geq c_{k_L}\}$, where $\sigma(1, \cdots, L)$ denotes the set of all possible permutation of $(1, \cdots, L)$. It's easy to see that $\cup_{l=1}^{L} \cup_{(k_1, \cdots, k_L) \in \sigma(1, \cdots, L)} \mathcal{C}_{lk_1 \cdots k_L} = \mathbb{R}^L$ Then, consider function $g(\boldsymbol{c}) = |\sum_{l=1}^{L} c_l n_l s_l| - \lambda_1 \sum_{l=1}^{L} |c_l| - \lambda_2 \sum_{l \sim l'} |c_l - c_{l'}|$. Note that, $g(\boldsymbol{c})$ over each region $\mathcal{C}_{lk_1 \cdots k_L}$ is a convex function. By the maximal principle, its maximum (over each region) can be attained at the extreme points of $\mathcal{C}_{lk_1 \cdots k_L}$. It is easy to show that the extreme points must be of the form $\boldsymbol{c} = (t\mathbf{1}_{\mathcal{I}}, \mathbf{0}_{\mathcal{I}^c})$ for some $\mathcal{I} \subseteq V$ and $t \neq 0$, that is the non-zero components must to equal to each other. Hence, $g(\boldsymbol{c})$ evaluated at the extreme points of $\mathcal{C}_{lk_1 \cdots k_L}$ reduces to $|\sum_{l \in \mathcal{I}} n_l s_l| - \lambda_1 |\mathcal{I}| - \lambda_2 d(\mathcal{I}, \mathcal{I}^c)$ for some $\mathcal{I} \subseteq V$, which, by assumption, is always nonpositive. This completes the proof. □

Proof of Theorem 2: We shall use the KKT condition of (12), or local optimality, which is in the form of

$$n_l\hat{\boldsymbol{\Omega}}_l^{-1} + n_l\boldsymbol{S}_l + \lambda_1\partial\|\hat{\boldsymbol{\Omega}}_l\|_{1,\text{off}} + \lambda_2\sum_{l':l\sim l'}\partial\|\hat{\boldsymbol{\Omega}}_l - \hat{\boldsymbol{\Omega}}_{l'}\|_{1,\text{off}} = \mathbf{0}, \quad l = 1,\cdots,L, \tag{28}$$

where $\hat{\boldsymbol{\Omega}}_l^{-1}$ is the inversion of matrices $\hat{\boldsymbol{\Omega}}_l$ and $\partial\|\cdot\|_1$ denotes the subgradient of the ℓ_1 function. If $\hat{\boldsymbol{\omega}}_{jkl} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$, then $\left(\hat{\boldsymbol{\Omega}}_l^{-1}\right)_{jk} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$. By Lemma 2, we must have $(s_{jk1},\cdots,s_{jkL}) \in \mathcal{S}$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c$. Conversely, if $(s_{jk1},\cdots,s_{jkL}) \in \mathcal{S}$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c$, again by Lemma 2, the KKT condition in (28) holds at $\hat{\boldsymbol{\omega}}_{jkl} = 0, l = 1,\cdots,L$ for jkth components for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$. Hence, $\hat{\boldsymbol{\omega}}_{jkl} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$. This completes the proof. □

Proof of Theorem 3: Let $(\hat{\boldsymbol{\Omega}}_1^{(m)},\cdots,\hat{\boldsymbol{\Omega}}_L^{(m)})$ be the DC solution at iteration m. If the diagonal matrix is initialized as in **Algorithm 1**, then an application of **Theorem 2** on $(\hat{\boldsymbol{\Omega}}_1^{(1)},\cdots,\hat{\boldsymbol{\Omega}}_L^{(1)})$ yields that $(s_{jk1},\cdots,s_{jkL}) \in \mathcal{S}$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c$, implying that $\hat{\omega}_{jkl}^{(1)} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$. Next, we prove by induction that if $(s_{jk1},\cdots,s_{jkL}) \in \mathcal{S}$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c$, then $\hat{\omega}_{jkl}^{(m)} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$ holds for any $m \ge 1$. Suppose that $\hat{\omega}_{jkl}^{(m-1)} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$ holds for some $m \ge 2$, then at DC iteration m, $|\hat{\omega}_{jkl}^{(m-1)}| = 0 \le \tau, |\hat{\omega}_{jkl}^{(m-1)} - \hat{\omega}_{jkl'}^{(m-1)}| = 0 \le \tau$. This, together with **Theorem 2**, again implies that $\hat{\omega}_{jkl}^{(m)} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$. Using the finite convergence of the DC algorithm, c.f., **Theorem 1**, we have $(s_{jk1},\cdots,s_{jkL}) \in \mathcal{S}$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c$, implying that $\hat{\omega}_{jkl}^{dc} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$. Conversely, if for some $\mathcal{J}$ $\hat{\omega}_{jkl}^{dc} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$, consider the next DC iteration, we have $\hat{\omega}_{jkl}^{m*+1} = 0$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c, 1 \le l \le L$. Using the same argument as above with the converse part of **Theorem 2**, we obtain that $(s_{jk1},\cdots,s_{jkL}) \in \mathcal{S}$ for any $j \in \mathcal{J}, k \in \mathcal{J}^c$. This completes the proof. □

Proof of Corollary 1: For the fused graph, let $\mathcal{I} = \{1,\cdots,l\}$, $\{L-l+1,\cdots,L\}$, $\{l_1+1,\cdots,l_2\}$ and $\{1,\cdots,L\}$, then if $|\sum_{l\in\mathcal{I}} n_l s_l| \le \lambda_1|\mathcal{I}| + \lambda_2 d(\mathcal{I},\mathcal{I}^c)$ then $\left|\sum_{i=1}^{l} n_i s_i\right| \le l\lambda_1+\lambda_2$, $\left|\sum_{i=L-l+1}^{L} n_i s_i\right| \le l\lambda_1+\lambda_2$, $\left|\sum_{i=l_1+1}^{l_2} n_i s_i\right| \le (l_2-l_1)\lambda_1+2\lambda_2$, $\left|\sum_{i=1}^{L} n_i s_i\right| \le L\lambda_1$. Conversely, if $\mathcal{I} = \{1,\cdots,L\}$, then $\left|\sum_{i\in\mathcal{I}} n_i s_i\right| \le \lambda_1|\mathcal{I}| + \lambda_2 d(\mathcal{I},\mathcal{I}^c) = L\lambda_1$. Next, assume that $\mathcal{I} \ne \{1,\cdots,L\}$, and write $\mathcal{I} = \cup_{k=1}^{q}\{i_k, i_k+1,\cdots,i_k+l_k\}$ with $i_1 \le i_1+l_1 < i_2 < i_2+l_2 < \cdots <$

$i_q < i_q + l_q$. Then

$$\begin{aligned}\Big|\sum_{i\in\mathcal{I}} n_i s_i\Big| &\le \sum_{k=1}^{q}\Big|\sum_{i=i_k}^{i_k+l_k} n_i s_i\Big| \le \lambda_1\sum_{k=1}^{q} l_k + 2(q-2)\lambda_2 + \big(\mathbb{I}(i_1\neq 1)+\mathbb{I}(i_q+l_q\neq L)+2\big)\lambda_2 \\ &= |\mathcal{I}|\lambda_1 + 2(q-1)\lambda_2 + \big(\mathbb{I}(i_1\neq 1)+\mathbb{I}(i_q+l_q\neq L)\big)\lambda_2 = |\mathcal{I}|\lambda_1 + d(\mathcal{I},\mathcal{I}^c)\lambda_2.\end{aligned}$$

In the case of the complete graph, set $\mathcal{I} = \{k_1,\cdots,k_l\}$, $\{k_{L-l+1},\cdots,k_L\}$, given $s_{k_1}\le\cdots\le s_{k_L}$, then we have $\Big|\sum_{i=1}^{l} n_{k_i}s_{k_i}\Big| \le l\lambda_1 + l(L-l)\lambda_2$, $\Big|\sum_{i=L-l+1}^{L} n_{k_i}s_{k_i}\Big| \le l\lambda_1 + l(L-l)\lambda_2$. Conversely, for any $\mathcal{I}$, $\Big|\sum_{i\in\mathcal{I}} n_i s_i\Big| \le \max\Big(\Big|\sum_{i=1}^{|\mathcal{I}|} n_{k_i}s_{k_i}\Big|, \Big|\sum_{i=L-|\mathcal{I}|+1}^{L} n_{k_i}s_{k_i}\Big|\Big) \le l\lambda_1 + l(L-l)\lambda_2$. This completes the proof. □.

Proof of Theorem 4: The proof uses a large deviation probability inequality of [23] to treat one-sided log-likelihood ratios with constraints. This enables us to obtain sharp results without a moment condition on both tails of the log-likelihood ratios.

Recall that $\mathcal{S} = \big\{\boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{\eta}) : |A(\boldsymbol{\beta})| \le d_0, C(\boldsymbol{\beta},\mathcal{E}) \le c_0, \mathcal{G}(\boldsymbol{\beta}) \neq \mathcal{G}(\boldsymbol{\beta}^0)\big\}$ and $\mathcal{S}_A = \big\{\boldsymbol{\theta}\in\mathcal{S} : A(\boldsymbol{\beta}) = A\big\}$. Let a class of candidate subsets be $\mathcal{A} \equiv \{A \neq A^0 : |A| \le d_0\}$ for sparseness pursuit. Note that any $A \subset \{1,\cdots,d\}$ can be partitioned into $(A\setminus A^0)\cup(A\cap A^0)$. Then we partition $\mathcal{S}$ accordingly with $\mathcal{S} = \cup_{i=0}^{d_0}\cup_{A\in\mathcal{B}_i}\mathcal{S}_A$, where $\mathcal{B}_i = \mathcal{A}\cap\{A : |A^0\setminus A| = i\}$, with $|\mathcal{B}_i| = \binom{d_0}{d_0-i}\sum_{j=0}^{i}\binom{d-d_0}{j}, i = 0,\cdots,d_0$. Moreover, $\mathcal{S}_A = \cup_{\mathcal{G}\in\{\mathcal{G}(\boldsymbol{\beta}):\boldsymbol{\theta}=(\boldsymbol{\beta},\boldsymbol{\eta})\in\mathcal{S}_A\}}\mathcal{S}_{\mathcal{G}}$, where $\mathcal{S}_{\mathcal{G}} = \big\{\boldsymbol{\theta} = (\boldsymbol{\beta},\boldsymbol{\eta})\in\mathcal{S} : \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}\big\}$. So $\mathcal{S} = \cup_{i=0}^{d_0}\cup_{A\in\mathcal{B}_i}\cup_{\mathcal{G}\in\{\mathcal{G}(\boldsymbol{\beta}):\boldsymbol{\theta}=(\boldsymbol{\beta},\boldsymbol{\eta})\in\mathcal{S}_A\}}\mathcal{S}_{\mathcal{G}}$.

To bound the error probability, note that if $\hat{\mathcal{G}}^{l_0} = \mathcal{G}^0$ then $\hat{\boldsymbol{\theta}}^{\ell_0} = \hat{\boldsymbol{\theta}}^o$ then $\hat{\boldsymbol{\theta}}^{\ell_0} = \hat{\boldsymbol{\theta}}^o$, by Definition 1. Conversely, if $\hat{\boldsymbol{\theta}}^{\ell_0} = \hat{\boldsymbol{\theta}}^o$ or $\hat{\boldsymbol{\beta}}^{\ell_0} = \hat{\boldsymbol{\beta}}^o$, then $\hat{\mathcal{G}}^{l_0} = \mathcal{G}^0$. Thus $\{\hat{\mathcal{G}}^{l_0} = \mathcal{G}^0\}\{\hat{\boldsymbol{\theta}}^{\ell_0} = \hat{\boldsymbol{\theta}}^o\}$. So $\{\hat{\boldsymbol{\theta}}^{\ell_0}\neq\hat{\boldsymbol{\theta}}^o\}\subseteq\{L(\hat{\boldsymbol{\theta}}^{\ell_0}) - L(\hat{\boldsymbol{\theta}}^o)\ge 0\}\subseteq\{l(\hat{\boldsymbol{\theta}}^{\ell_0}) - l(\boldsymbol{\theta}^0)\ge 0\}$. This together with $\{\hat{\boldsymbol{\theta}}^{\ell_0}\neq\hat{\boldsymbol{\theta}}^o\}\subseteq\{\hat{\boldsymbol{\theta}}^{\ell_0}\in\mathcal{S}\}$ implies that $\{\hat{\boldsymbol{\theta}}^{\ell_0}\neq\hat{\boldsymbol{\theta}}^o\}\subseteq\{l(\hat{\boldsymbol{\theta}}^{\ell_0}) - l(\boldsymbol{\theta}^0)\ge 0\}\cap\{\hat{\boldsymbol{\theta}}^{\ell_0}\in\mathcal{S}\}$. Consequently, $I \equiv \mathbb{P}\big(\hat{\boldsymbol{\theta}}^{\ell_0}\neq\hat{\boldsymbol{\theta}}^o\big) \le \mathbb{P}\Big(L(\hat{\boldsymbol{\theta}}^{\ell_0}) - L(\boldsymbol{\theta}^0)\ge 0; \hat{\boldsymbol{\theta}}^{\ell_0}\in\mathcal{S}\Big)$ is upper bounded by

$$\begin{aligned}&\sum_{i=0}^{d_0}\sum_{A\in\mathcal{B}_i}\sum_{\mathcal{G}\in\{\mathcal{G}(\boldsymbol{\beta}):\boldsymbol{\theta}\in\mathcal{S}_A\}}\mathbb{P}^*\Big(\sup_{\boldsymbol{\theta}\in\mathcal{S}_{\mathcal{G}}}\big(L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}^0)\big)\ge 0\Big) \\ \le\ &\sum_{i=0}^{d_0}\sum_{A\in\mathcal{B}_i}\sum_{\mathcal{G}\in\{\mathcal{G}(\boldsymbol{\beta}):\boldsymbol{\theta}\in\mathcal{S}_A\}}\mathbb{P}^*\Big(\sup_{\left\{-\log(1-h^2(\boldsymbol{\theta},\boldsymbol{\theta}^0))\ge\max(i,1)C_{\min}(\boldsymbol{\theta}^0),\ \boldsymbol{\theta}\in\mathcal{S}_{\mathcal{G}}\right\}}\big(L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}^0)\big)\ge 0\Big),\end{aligned}$$

where $\mathbb{P}^*$ is the outer measure and the last two inequalities use the fact that $\mathcal{S} = \cup_{i=0}^{d_0}\cup_{A\in\mathcal{B}_i}$

$\cup_{\mathcal{G}\in\{\mathcal{G}(\boldsymbol{\beta}):\boldsymbol{\theta}=(\boldsymbol{\beta},\boldsymbol{\eta})\in\mathcal{S}_A\}}\mathcal{S}_{\mathcal{G}}$ and $\mathcal{S}_{\mathcal{G}} \subseteq \{\boldsymbol{\theta} : \max(|A^0 \setminus A|, 1)C_{\min}(\boldsymbol{\theta}^0) \leq -\log(1 - h^2(\boldsymbol{\theta}, \boldsymbol{\theta}^0))\}$ for $\mathcal{G} \in \{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta}) \in \mathcal{S}_A\}$.

For I, we apply Theorem 1 of [23] to bound each term. Towards this end, we verify their entropy condition (3.1) for the local entropy over $\mathcal{S}_{\mathcal{G}}$ for $\mathcal{G} \in \{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta}) \in \mathcal{S}_A\}$, $A \in \mathcal{B}_i$ and $i = 0, \cdots, d_0$. Under **Assumption A** $\varepsilon = \varepsilon_{n,p_0,p} = (2c_0)^{1/2}c_4^{-1}\log(2^{1/2}/c_3)\log p(\frac{p_0}{n})^{1/2}$ satisfies there with respect to $\varepsilon > 0$, that is,

$$\sup_{\{0\leq|A|\leq p_0\}} \int_{2^{-8}\varepsilon^2}^{2^{1/2}\varepsilon} H^{1/2}(t/c_3, \mathcal{B}_A)dt \leq p_0^{1/2}2^{1/2}\varepsilon\log(2/2^{1/2}c_3) \leq c_4 n^{1/2}\varepsilon^2. \tag{29}$$

for some constant $c_3 > 0$ and c_4, say $c_3 = 10$ and $c_4 = \frac{(2/3)^{5/2}}{512}$. By **Assumption A**, $C_{\min}(\boldsymbol{\theta}^0) \geq \varepsilon_{n,p_0,p}^2$ implies (29), provided that $d_0 \geq (2c_0)^{1/2}c_4^{-1}\log(2^{1/2}/c_3)$.

Now, let $S_i^\star = \max_{A\in\mathcal{B}_i} \#\{\mathcal{G} : \mathcal{I}_0^c = A\}$ and $\log(S^\star) = \max_{1\leq i\leq p_0}\log(S_i^\star)/i$. Using inequalities for binomial coefficients: $\sum_{j=0}^i \binom{d-d_0}{j} \leq (d-d_0)^i$ and $\binom{d_0}{i} \leq d_0^i$, $|\mathcal{B}_i| = \binom{d_0}{d_0-i}\sum_{j=0}^i \binom{d-d_0}{j} \leq \big(d(d-d_0)\big)^i \leq (d^2/4)^i$, we have, by Theorem 1 of [23], that for a constant $c_2 > 0$, say $c_2 = \frac{4}{27}\frac{1}{1926}$,

$$\begin{aligned} I &\leq \sum_{i=0}^{d_0} |\mathcal{B}_i| S_i^\star \exp\big(-c_2 n i C_{\min}(\boldsymbol{\theta}^0)\big) \leq \sum_{i=0}^{d_0} \big(\frac{d^2}{4}\big)^i S_i^\star \exp\big(-c_2 n i C_{\min}(\boldsymbol{\theta}^0)\big) \\ &\leq \exp\big(-c_2 n C_{\min}(\boldsymbol{\theta}^0) + 2\log d + \log(S^\star)\big). \end{aligned}$$

This completes the proof. □

Proof of Theorem 5: The proof is similar to that of Theorem 4 with some minor modifications. Given τ and $\boldsymbol{\beta} \in \mathbb{R}^d$, a partition $\mathcal{G}^\tau(\boldsymbol{\beta}) = (\mathcal{I}_0(\boldsymbol{\beta}), \cdots, \mathcal{I}_{K(\boldsymbol{\beta})}(\boldsymbol{\beta}))$ associated with $\boldsymbol{\beta}$ is defined to satisfy the following (i) $\max_{j\in\mathcal{I}_0(\boldsymbol{\beta})}|\boldsymbol{\beta}_j| \leq \tau$; (ii) $\big|\beta_{j_1} - \beta_{j_2}\big| \leq \tau$ for any j_1, j_2 in different groups. Let $A^\tau(\boldsymbol{\beta}) = \mathcal{I} \setminus \mathcal{I}_0(\boldsymbol{\beta})$.

The rest of the proof is basically the same as that in Theorem 2 with a modification that $\mathcal{G}(\boldsymbol{\beta})$ and $A(\boldsymbol{\beta})$ are replaced by $\mathcal{G}^\tau(\boldsymbol{\beta})$ and $A^\tau(\boldsymbol{\beta})$ respectively. Here, $\mathcal{S} = \big\{\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta}) : \sum_{j=1}^p J_\tau(|\beta_j|) \leq d_0, \sum_{(jj')\in\mathcal{E}} J_\tau\big(\big|\beta_j - \beta_{j'}\big|\big) \leq c_0, \mathcal{G}^\tau(\boldsymbol{\beta}) \neq \mathcal{G}(\boldsymbol{\beta}^0)\big\}$, $C^\tau(\boldsymbol{\beta}, \mathcal{E}) = \sum_{(jj')\in\mathcal{E}} I(|\beta_j - \beta_{j'}| \neq 0)$, $\mathcal{S}_A = \big\{\boldsymbol{\theta} \in \mathcal{S} : A^\tau(\boldsymbol{\beta}) = A\big\}$ and $\mathcal{S}_{\mathcal{G}} = \big\{\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta}) \in \mathcal{S} : \mathcal{G}^\tau(\boldsymbol{\beta}) = \mathcal{G}\big\}$.

Next, we show that $\hat{\boldsymbol{\theta}}^g = \hat{\boldsymbol{\theta}}^o$ if and only if $\hat{\mathcal{G}}^g = \mathcal{G}^0$, where $\hat{\mathcal{G}}^g \equiv \mathcal{G}^\tau(\hat{\boldsymbol{\beta}}^g)$. Now $d_1 \equiv$

$|\mathcal{I} \setminus \hat{\mathcal{I}}_0^0| = d_0$. By (14), $\frac{1}{\tau}\sum_{j\in\hat{\mathcal{I}}_0}|\hat{\beta}_j^g| + d_1 \le d_0$, with $d_0 = d_1$, yields that $\hat{\beta}_j^g = 0$; $j \in \mathcal{I}_0^1$. In addition, the second constraint of (14) implies $\sum_{i=1}^K \sum_{jj'\in\mathcal{I}_i,(jj')\in\mathcal{E}} \frac{|\hat{\beta}_j^g - \hat{\beta}_{j'}^g|}{\tau} \le 0$, yielding that $\hat{\beta}_{j_1}^g = \hat{\beta}_{j_1}^g$ for any $j_1, j_2 \in \hat{\mathcal{I}}_i, (j_1, j_2) \in \mathcal{E}$, $i = 1, \cdots, K$. By graph consistency of $\mathcal{U}$, $\mathcal{U}$ is connected over $\hat{\mathcal{I}}_i$, implying that $\hat{\beta}_{j_1}^g = \hat{\beta}_{j_1}^g$ for any $j_1, j_2 \in \hat{\mathcal{I}}_i$, $i = 1, \cdots, K$. This further implies that $\hat{\boldsymbol{\beta}}^g = \hat{\boldsymbol{\beta}}^o$ and $\hat{\boldsymbol{\theta}}^g = \hat{\boldsymbol{\theta}}^o$, meaning that that $\{\widehat{\mathcal{G}}^g = \mathcal{G}^0\} \subseteq \{\hat{\boldsymbol{\theta}}^g = \hat{\boldsymbol{\theta}}^o\}$. On the other hand, it is obvious that if $\hat{\boldsymbol{\theta}}^g = \hat{\boldsymbol{\theta}}^o$ then $\{\widehat{\mathcal{G}}^g = \mathcal{G}^0\}$. Hence, $\{\widehat{\mathcal{G}}^g = \mathcal{G}^0\} = \{\hat{\boldsymbol{\theta}}^g = \hat{\boldsymbol{\theta}}^o\}$ from which we conclude that $\{\hat{\boldsymbol{\theta}}^g \neq \hat{\boldsymbol{\theta}}^o\} \subseteq \{\hat{\boldsymbol{\theta}}^g \in \mathcal{S}\}$. This together with $\{\hat{\boldsymbol{\theta}}^g \neq \hat{\boldsymbol{\theta}}^o\} \subseteq \{L(\hat{\boldsymbol{\theta}}^g) - L(\boldsymbol{\theta}^o) \ge 0\} \subseteq \{L(\hat{\boldsymbol{\theta}}^g) - L(\boldsymbol{\theta}^0) \ge 0\}$ implies that $\mathbb{P}\big(\hat{\boldsymbol{\theta}}^g \neq \hat{\boldsymbol{\theta}}^o\big) \le \mathbb{P}\Big(L(\hat{\boldsymbol{\theta}}^g) - L(\boldsymbol{\theta}^0) \ge 0; \hat{\boldsymbol{\theta}}^g \in \mathcal{S}\Big)$ is bounded by

$$
\begin{aligned}
&\sum_{i=0}^{d_0}\sum_{A\in\mathcal{B}_i}\sum_{\mathcal{G}\in\{\mathcal{G}^\tau(\boldsymbol{\beta}):\boldsymbol{\theta}=(\boldsymbol{\beta},\boldsymbol{\eta})\in\mathcal{S}_A\}} \mathbb{P}^*\Big(\sup_{\boldsymbol{\theta}\in\mathcal{S}_\mathcal{G}}\big(L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}^0)\big) \ge 0\Big) \\
\le\ &\sum_{i=0}^{d_0}\sum_{A\in\mathcal{B}_i}\sum_{\mathcal{G}\in\{\mathcal{G}^\tau(\boldsymbol{\beta}):\boldsymbol{\theta}\in\mathcal{S}_A\}} \mathbb{P}^*\Big(\sup_{\left\{-\log(1-h^2(\boldsymbol{\theta},\boldsymbol{\theta}^0))\ge d_1\max(i,1)C_{\min}(\boldsymbol{\theta}^0) - d_3\tau^{d_2}d, \boldsymbol{\theta}\in\mathcal{S}_\mathcal{G}\right\}} \\
&\big(L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}^0)\big) \ge 0\Big),
\end{aligned}
$$

where the last step uses the fact that $\{\boldsymbol{\theta} \in \mathcal{S}_\mathcal{G}\} \subseteq \{-\log(1-h^2(\boldsymbol{\theta}^\tau, \boldsymbol{\theta}^0)) \ge \max(i,1)C_{\min}(\boldsymbol{\theta}^0)\}$ $\subseteq \{-\log(1-h^2(\boldsymbol{\theta}, \boldsymbol{\theta}^0)) \ge d_1\max(i,1)C_{\min}(\boldsymbol{\theta}^0) - d_3\tau^{d_2}d\}$, under **Assumption B**. Then, for some constant c_3, $\mathbb{P}\big(\hat{\boldsymbol{\theta}}^g \neq \hat{\boldsymbol{\theta}}^o\big)$ is upper bounded by

$$
\begin{aligned}
&\sum_{i=0}^{d_0}|\mathcal{B}_i|S_i^\star \exp\big(-c_3 n i C_{\min}(\boldsymbol{\theta}^0)\big) \le \sum_{i=0}^{d_0}\big(\frac{d^2}{4}\big)^i S_i^\star \exp\big(-c_3 n i C_{\min}(\boldsymbol{\theta}^0)\big) \\
\le\ &\exp\big(-c_3 n C_{\min}(\boldsymbol{\theta}^0) + 2\log d + \log(S^\star)\big),
\end{aligned}
$$

provided that $\tau \le \big(\frac{(d_1-c_3)C_{\min}(\boldsymbol{\theta}^0)}{d_3 d}\big)^{1/d_2}$. This completes the proof. □

Proof of Corollary 2: First we derive an upper bound of S^*. Let p_l^0 the number of nonzero elements of the precision matrix in the sth cluster for $l = 1, \cdots, L$. Let $p_0 = d_0/L = \frac{p_1^0 + \cdots + p_L^0}{L}$ be the average number of nonzero elements. For any $\boldsymbol{\theta} \in \mathcal{S}_A$ with $|A^0 \setminus A| = i$, let $\mathcal{Q} = \{(j,k) : j > k, \exists l, x_{jul} \neq 0\}$ and $|\mathcal{Q}| = q_0$. Let $a_{jk} = \#\{l : x_{jkl} \neq 0\}$ for $(j,k) \in \mathcal{Q}$. Note that $q_0 \le p^0$ and $\sum_{(j,k)\in\mathcal{Q}} a_{jk} \le d_0$ since $|A| \le d_0$. By the definition of S_i^*, we have

$$S_i^* \;\le\; \sum_{\sum_{(j,k)\in\mathcal{Q}} r_{jk}\le g_0} \prod_{(j,k)\in\mathcal{Q}} \binom{a_{jk}-1}{r_{jk}} = \sum_{g=0}^{g_0} \binom{\sum_{(j,k)\in\mathcal{Q}} a_{jk} - q_0}{g}$$

$$\le \sum_{g=0}^{g_0} \binom{d_0-p_0}{g} \le (g_0+1)\big(e\frac{d_0-p_0}{g_0}\big)^{g_0}. \tag{30}$$

This together with $\log(1+g_0) \le g_0$ implies $\log S^* \le 2g_0 \max(\log(d_0/g_0), 1)$. To lower bound $C_{min}(\boldsymbol{\theta}^0)$, we proceed similarly with the proof of **Proposition 2** in [17]. Specifically, note that $h^2(\boldsymbol{\theta}, \boldsymbol{\theta}^0) = 1 - \prod_{l=1}^L \big(1 - h^2(\boldsymbol{\Omega}_l, \boldsymbol{\Omega}_l^0)\big)$. Thus,

$$-\log(1-h^2(\boldsymbol{\theta},\boldsymbol{\theta}^0)) \;=\; \sum_{l=1}^L \Big(-\log\big(1-h^2(\boldsymbol{\Omega}_l,\boldsymbol{\Omega}_l^0)\big)\Big). \tag{31}$$

An application of Proposition 2 of [17] yields that each term in (31) is lower bounded by $c^\star \|\boldsymbol{\Omega}_l - \boldsymbol{\Omega}_l^0\|_2^2$. Therefore, $-\log(1-h^2(\boldsymbol{\theta},\boldsymbol{\theta}^0)) \ge c^\star \min_{1\le l\le L} c_{min}\big(H_l\big) \sum_{l=1}^L \|\boldsymbol{\Omega}_l - \boldsymbol{\Omega}_l^0\|_2^2$. Now if $A_0 \setminus A \neq \emptyset$, we have $\sum_{l=1}^L \|\boldsymbol{\Omega}_l - \boldsymbol{\Omega}_l^0\|_2^2 \ge |A_0 \setminus A| \min_{(j,k,l):\omega_{jkl}\neq 0} \omega_{jkl}^2$. If $A_0 \setminus A = \emptyset$, then by definition of $\mathcal{S}$, there must exist (j,k,l) such that $\omega_{jkl} = \omega_{jk(l+1)}$ and $\omega_{jkl}^0 \neq \omega_{jk(l+1)}^0$. Here

$$\sum_{l=1}^L \|\boldsymbol{\Omega}_l - \boldsymbol{\Omega}_l^0\|_2^2 \;\ge\; (\omega_{jkl}-\omega_{jkl}^0)^2 + (\omega_{jk(l+1)} - \omega_{jk(l+1)}^0)^2 \ge \frac{1}{2}(\omega_{jkl}^0 - \omega_{jk(l+1)}^0)^2$$

$$\ge \frac{1}{2} \min_{\{(j,k,l):\ \omega_{jkl}^0 \neq \omega_{jk(l+1)}^0\}} \big(\omega_{jkl}^0 - \omega_{jk(l+1)}^0\big)^2.$$

A combination of both the cases yield that

$$-\log(1-h^2(\boldsymbol{\theta},\boldsymbol{\theta}^0))/\max\big(|A_0\setminus A|, 1\big) \ge c^\star \min_{1\le l\le L} c_{min}\big(H_l\big)\eta_{min}^2.$$

which, after taking infimum over $\mathcal{S}$, leads to $C_{min}(\boldsymbol{\theta}^0) \ge c^\star c_{min}\big(H_l\big)\eta_{min}^2$. This, together with, the upper bound on $\log S^\star$ in Theorems 4 and 5, gives a sufficient condition for simultaneous pursuit of sparseness and clustering: $\min_{1\le l\le L} c_{min}\big(H_l\big)\eta_{min}^2 \ge c_0 \frac{\log(p^2L) - g_0 \max(\log(d_0/g_0),1)}{n}$, for some $c_0 > 0$. Moreover, under this condition, $\mathbb{P}\Big(\hat{\boldsymbol{\Omega}}^{\ell_0} \neq \hat{\boldsymbol{\Omega}}^o\Big)$ and $\mathbb{P}\Big(\hat{\boldsymbol{\Omega}}^g \neq \hat{\boldsymbol{\Omega}}^o\Big) \to 0$ as $n, d \to +\infty$. This completes the proof. □

References

[1] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine*

Learning Research, 9:485–516, 2008.

[2] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.

[3] T.H. Cormen. *Introduction to algorithms*. The MIT press, 2001.

[4] J. Fan, Y. Feng, and Y. Wu. Network exploration via the adaptive lasso and scad penalties. *The annals of applied statistics*, 3(2):521–541, 2009.

[5] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432, 2008.

[6] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1, 2011.

[7] M. Kolar and E.P. Xing. On time varying undirected graphs. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011.

[8] A.N. Kolmogorov and V.M. Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

[9] Steffen L Lauritzen. *Graphical models*. Oxford University Press, 1996.

[10] B. Li, H. Chun, and H. Zhao. Sparse estimation of conditional graphical models with application to gene networks. *Journal of the American Statistical Association*, 107:152–167, 2012.

[11] H. Li and J. Gui. Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302, 2006.

[12] R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *The Journal of Machine Learning Research*, 13:781–794, 2012.

[13] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

[14] G.V. Rocha, P. Zhao, and B. Yu. A path following algorithm for sparse pseudo-likelihood inverse covariance estimation (splice). *Arxiv preprint arXiv:0807.3734*, 2008.

[15] A.J. Rothman, P.J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

[16] K. Sachs, O. Perez, D. Pe'er, D.A. Lauffenburger, and G.P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523, 2005.

[17] X. Shen, W. Pan, and Y. Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of American Statistical Association*, 107:223–232, 2012.

[18] Teppei Shimamura, Seiya Imoto, Rui Yamaguchi, Masao Nagasaki, and Satoru Miyano. Inferring dynamic gene networks under varying conditions for transcriptomic network comparison. *Bioinformatics*, 26(8):1064–1072, 2010.

[19] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67:91–108, 2005.

[20] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[21] R. Verhaak, K.A Hoadley, E. Purdom, V. Wang, Y. Qi, M. D Wilkerson, C. R. Miller, L. Ding, T. Golub, J.P. Mesirov, et al. An integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr and nf1. *Cancer cell*, 17(1):98, 2010.

[22] D. M. Witten, J. H. Friedman, and N. Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20:892–900, 2011.

[23] W.H. Wong and X. Shen. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *The Annals of Statistics*, pages 339–362, 1995.

[24] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19, 2007.

[25] S. Zhou, J. Lafferty, and L. Wasserman. Time varying undirected graphs. *Machine Learning*, 80(2):295–319, 2010.

[26] Y. Zhu, X. Shen, and W. Pan. Simultaneous grouping pursuit and feature selection over an undirected graph. *Journal of the American Statistical Association*, 108:713–725, 2013.

Table 1: Average entropy loss, denoted by EL, (SD in parentheses) and average quadratic loss, denoted by QL, (SD in parentheses), average false positive for sparseness pursuit, denote by FPV, (SD in parentheses), average false negative for sparseness pursuit, denoted as FNV, (SD in parentheses), average false positive for grouping, denoted by FPG, (SD in parentheses), and average false negative for grouping, denoted by FNG, (SD in parentheses), based on 100 simulations, for estimating precision matrices in Example 1 with $n = 120$. Here "Smooth", "Lasso", "TLP", "Our-con" and "Ours" denote estimation of individual matrices with kernel smoothing method proposed in [25], the L_1 sparseness penalty, that with non-convex TLP penalty, the convex counterpart of our method with the L_1-penalty for sparseness and clustering, and our non-convex estimates by solving (2) with penalty (6). The best performer is bold-faced.

(p, L)	Method	EL	QL	FPV	FNV	FPG	FNG
$(30, 4)$	Smooth	0.570(.005)	2.617(.266)	.312(.016)	.000(.000)	.392(.020)	.000(.000)
	Lasso	1.547(.074)	5.416(.393)	.200(.009)	.000(.000)	.377(.015)	.000(.000)
	TLP	0.746(.084)	4.688(.617)	.043(.006)	.001(.002)	.108(.008)	.000(.000)
	Our-con	1.288(.064)	3.700(.270)	.129(.016)	.000(.000)	.045(.020)	.251(.032)
	Ours	**0.525(.055)**	**3.494(.418)**	.040(.009)	.000(.000)	.009(.007)	.267(.043)
$(200, 4)$	Smooth	7.118(.173)	**22.45(2.61)**	.087(.017)	.000(.000)	.106(.018)	.000(.000)
	Lasso	36.48(.426)	69.21(1.97)	.013(.001)	.000(.000)	.027(.001)	.000(.000)
	TLP	5.305(.351)	33.67(2.22)	.004(.000)	.005(.003)	.014(.000)	.000(.000)
	Our-con	36.28(.422)	66.53(1.87)	.010(.001)	.000(.000)	.012(.001)	.122(.021)
	Ours	**3.500(.164)**	23.71(1.33)	.003(.000)	.000(.000)	.001(.000)	.280(.007)
$(20, 30)$	Smooth	1.122(.023)	1.983(.056)	.131(.006)	.000(.000)	.223(.006)	.000(.000)
	Lasso	1.685(.028)	3.770(.113)	.152(.005)	.000(.000)	.314(.008)	.000(.000)
	TLP	0.507(.023)	3.081(.180)	.077(.004)	.000(.000)	.198(.005)	.000(.000)
	Ours-con	1.593(.028)	3.256(.097)	.136(.007)	.000(.000)	.055(.003)	.024(.004)
	Ours	**0.236(.015)**	**1.812(.130)**	.068(.016)	.000(.000)	.020(.002)	.032(.004)
$(10, 90)$	Smooth	0.339(.007)	0.603(.017)	.271(.014)	.000(.000)	.420(.012)	.000(.000)
	Lasso	0.575(.010)	1.439(.038)	.273(.007)	.000(.000)	.541(.009)	.000(.000)
	TLP	0.250(.009)	1.404(.061)	.196(.006)	.000(.000)	.434(.008)	.000(.000)
	Our-con	0.519(.009)	1.190(.032)	.284(.018)	.000(.000)	.071(.005)	.008(.002)
	Ours	**0.100(.005)**	**0.748(.043)**	.017(.020)	.000(.000)	.028(.004)	.012(.002)

Table 2: Average entropy loss, denoted by EL, (SD in parentheses) and average quadratic loss, denoted by QL, (SD in parentheses), average false positive for sparseness pursuit, denote by FPV, (SD in parentheses), average false negative for sparseness pursuit, denoted as FNV, (SD in parentheses), average false positive for grouping, denoted by FPG, (SD in parentheses), and average false negative for grouping, denoted by FNG, (SD in parentheses), based on 100 simulations, for estimating precision matrices in Example 2 with $n = 300$. Here "Smooth", "Lasso", "TLP", "Our-con" and "Ours" denote estimation of individual matrices with kernel smoothing method proposed in [25], the L_1 sparseness penalty, that with non-convex TLP penalty, the convex counterpart of our method with the L_1-penalty for sparseness and clustering, and our non-convex estimates by solving (2) with penalty (6). The best performer is bold-faced.

(p, L)	Method	EL	QL	FPV	FNV	FPG	FNG
$(30, 4)$	Smooth	0.418(.025)	1.081(.072)	.387(.054)	.009(.008)	.543(.060)	.002(.002)
	Lasso	0.732(.041)	1.840(.122)	.229(.011)	.061(.013)	.466(.016)	.006(.005)
	TLP	0.772(.055)	2.290(.192)	.038(.005)	.184(.020)	.162(.010)	.037(.014)
	Our-con	0.591(.039)	1.373(.104)	.081(.019)	.033(.013)	.056(.035)	.146(.021)
	Ours	**0.359(.036)**	**1.012(.111)**	.044(.009)	.031(.015)	.004(.002)	.233(.012)
$(200, 4)$	Smooth	3.198(.069)	**7.823(.187)**	.129(.010)	.030(.006)	.161(.010)	.013(.002)
	Lasso	6.902(.140)	17.91(.435)	.049(.001)	.234(.010)	.110(.002)	.039(.004)
	TLP	8.350(.215)	23.71(.710)	.003(.001)	.493(.015)	.017(.001)	.116(.010)
	Our-con	6.151(.148)	15.58(.439)	.014(.005)	.198(.013)	.030(.010)	.142(.046)
	Ours	**2.977(.135)**	8.108(.399)	.001(.000)	.191(.010)	.001(.000)	.284(.012)
(20,30)	Smooth	0.409(.011)	0.839(.006)	.063(.007)	.024(.004)	.290(.007)	.005(.001)
	Lasso	0.470(.011)	1.157(.034)	.290(.006)	.034(.004)	.611(.008)	.001(.001)
	TLP	0.491(.017)	1.421(.057)	.081(.004)	.118(.007)	.341(.006)	.004(.001)
	Our-con	0.303(.010)	0.682(.025)	.071(.013)	.008(.002)	.044(.014)	.023(.002)
	Ours	**0.111(.006)**	**0.317(.019)**	.012(.005)	.006(.003)	.011(.002)	.032(.003)
$(10, 90)$	Smooth	0.123(.003)	0.248(.007)	.132(.013)	.008(.002)	.518(.008)	.001(.000)
	Lasso	0.170(.003)	0.419(.010)	.405(.010)	.007(.002)	.798(.008)	.000(.000)
	TLP	0.155(.005)	0.439(.014)	.175(.007)	.020(.003)	.609(.007)	.000(.000)
	Our-con	0.099(.008)	0.230(.007)	.135(.024)	.000(.000)	.043(.015)	.001(.001)
	Ours	**0.040(.002)**	**0.117(.005)**	.015(.014)	.000(.000)	.009(.001)	.001(.001)

Table 3: Average entropy loss, denoted by EL, (SD in parentheses) and average quadratic loss, denoted by QL, (SD in parentheses), based on 100 simulations, for estimating multiple precision matrices in Example 3. Here "Smooth", "Lasso", "TLP", "Our-con" and "Ours" denote estimation of individual matrices with kernel smoothing method proposed in [25], the L_1 sparseness penalty, that with non-convex TLP penalty, the convex counterpart of our method with the L_1-penalty for sparseness and clustering, and our non-convex estimates by solving (2) with penalty (6). The best performer is bold-faced.

Set-up	Method	$n = 120$		$n = 300$	
(p,L)		EL	QL	EL	QL
(30 , 4)	Smooth	**0.468(.042)**	**0.941(.097)**	**0.231(.056)**	**0.476(.034)**
	Lasso	1.158(.062)	2.534(.175)	0.736(.036)	1.434(.086)
	TLP	1.546(.100)	3.625(.262)	0.575(.045)	1.301(.121)
	Our-con	0.897(.066)	1.823(.166)	0.699(.038)	1.317(.085)
	Ours	0.501(.063)	1.143(.160)	0.247(.017)	0.524(.042)
(200, 4)	Smooth	6.882(.220)	13.26(.535)	2.578(.066)	4.843(.130)
	Lasso	10.37(.173)	21.92(.498)	5.449(.094)	10.84(.211)
	TLP	12.34(.202)	25.87(.560)	5.523(.153)	12.08(.365)
	Our-con	6.091(.199)	12.34(.484)	4.625(.098)	8.625(.209)
	Ours	**5.079(.265)**	**11.84(.658)**	**1.682(.038)**	**3.551(.096)**
(20 , 30)	Smooth	0.490(.021)	0.878(.042)	0.278(.008)	0.492(.015)
	Lasso	0.786(.020)	1.670(.052)	0.564(.012)	1.066(.027)
	TLP	0.987(.036)	2.454(.107)	0.355(.014)	0.819(.035)
	Our-con	0.653(.023)	1.281(.055)	0.528(.012)	0.959(.027)
	Ours	**0.317(.013)**	**0.730(.036)**	**0.183(.005)**	**0.391(.014)**
(10 , 90)	Smooth	0.230(.008)	0.409(.017)	0.115(.003)	0.203(.005)
	Lasso	0.318(.008)	0.694(.022)	0.205(.004)	0.398(.008)
	TLP	0.402(.012)	1.010(.043)	0.148(.004)	0.346(.011)
	Our-con	0.240(.008)	0.487(.019)	0.180(.004)	0.335(.007)
	Ours	**0.158(.005)**	**0.369(.014)**	**0.082(.002)**	**0.176(.005)**

Table 4: Average entropy loss, denoted by EL, (SD in parentheses) and average quadratic loss, denoted by QL, (SD in parentheses), average false positive for sparseness pursuit, denote by FPV, (SD in parentheses), average false negative for sparseness pursuit, denoted as FNV, (SD in parentheses), average false positive for grouping, denoted by FPG, (SD in parentheses), and average false negative for grouping, denoted by FNG, (SD in parentheses), based on 100 simulations, for estimating precision matrices in Example 4. Here "Lasso", "TLP", "Our-con" and "Ours" denote estimation of individual matrices with the L_1 sparseness penalty, that with non-convex TLP penalty, the convex counterpart of our method with the L_1-penalty for sparseness and clustering, and our non-convex estimates by solving (2) with penalty (6). The best performer is bold-faced.

(p, L)	Method	EL	QL	FPV	FNV	FPG	FNG
$(1000, 4)$	Lasso	378.5(2.09)	829.4(13.3)	.0005(.0000)	.0270(.0030)	.0020(.0000)	.0002(.0003)
	TLP	36.05(.1.78)	201.9(8.81)	.0004(.0000)	.0270(.0030)	.0020(.0000)	.0002(.0003)
	Our-con	377.3(2.07)	805.2(12.9)	.0004(.0000)	.0110(.0020)	.0017(.0000)	.0497(.0040)
	Ours	**26.8(1.35)**	**160.9(7.266)**	.0003(.0000)	.0130(.0020)	.0017(.0000)	.0267(.0027)
$(2000, 4)$	Lasso	225.6(.413)	358.1(1.13)	.0009(.0000)	.0000(.0000)	.0018(.0000)	.0000(.0000)
	TLP	9.160(.083)	54.17(.654)	.0007(.0000)	.0000(.0000)	.0015(.0000)	.0000(.0000)
	Our-con	225.6(.413)	358.1(1.12)	.0009(.0000)	.0000(.0000)	.0018(.0000)	.0000(.0000)
	Ours	**8.617(.081)**	**51.79(.657)**	.0006(.0000)	.0000(.0000)	.0005(.0000)	.1750(.0020)

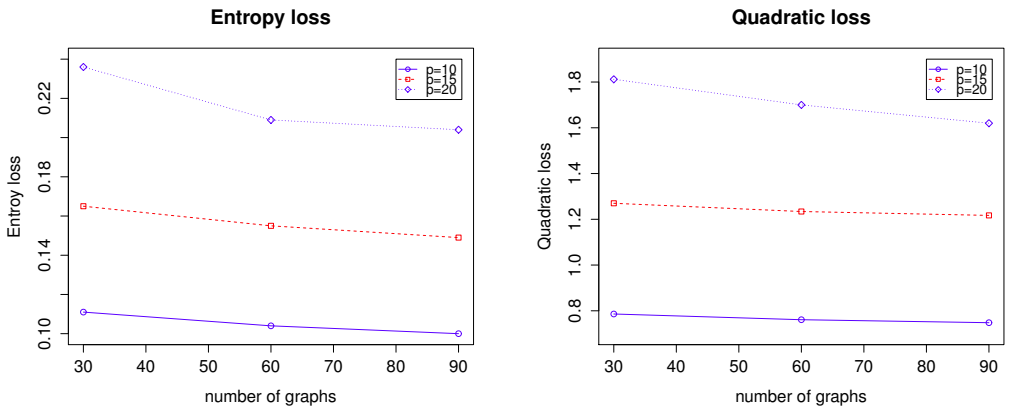
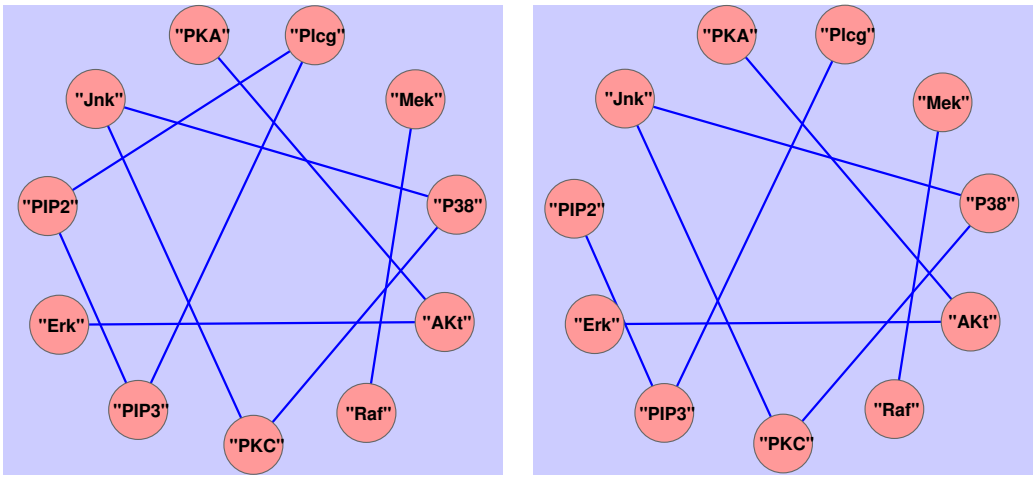


Figure 1: Average entropy and quadratic losses of the proposed method over different p and L values over 100 simulation replications in Example 1.



(a) Under the nine conditions. (b) Under the five conditions.

Figure 2: Reconstructed networks for simultaneous pursuit of clustering and sparsity.

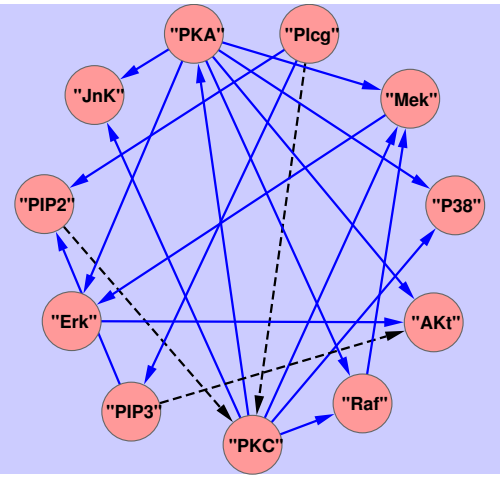


Figure 3: Signaling network reproduced from Figure 3(A) of [16], where the black dashed line represents links that have been missed by methods in [16].