IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 March 2025, accepted 8 April 2025, date of publication 11 April 2025, date of current version 23 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3559883

RESEARCH ARTICLE

Learning Multi-Attribute Differential Graphs With Non-Convex Penalties

JITENDRA K. TUGNAIT, (Life Fellow, IEEE)
Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA
e-mail: tugnajk@auburn.edu

This work was supported by the National Science Foundation under Grant CCF-2308473.

ABSTRACT We consider the problem of estimating differences in two multi-attribute Gaussian graphical models (GGMs) which are known to have similar structure, using a penalized D-trace loss function with non-convex penalties. The GGM structure is encoded in its precision (inverse covariance) matrix. Existing methods for multi-attribute differential graph estimation are based on a group lasso penalized loss function. In this paper, we consider a penalized D-trace loss function with non-convex [log-sum and smoothly clipped absolute deviation (SCAD)] penalties. Two proximal gradient descent methods are presented to optimize the objective function. Theoretical analysis establishing sufficient conditions for consistency in support recovery, convexity and estimation in high-dimensional settings is provided. We illustrate our approaches with numerical examples based on synthetic and real data.

INDEX TERMS Differential graph learning, undirected graph, multi-attribute graphs, log-sum, SCAD penalties.

I. INTRODUCTION

Graphical models are used to display and explore conditional independence structure of the random variables in a system [1], [2], [3]. The conditional statistical dependency structure among p random variables $x_1, x_2, \cdots, x_p$, is represented using an undirected graph $\mathcal{G} = (V, \mathcal{E})$ with a set of p vertices (nodes) $V = \{1, 2, \cdots, p\} = [p]$, and a corresponding set of (undirected) edges $\mathcal{E} \subseteq [p] \times [p]$. The graph $\mathcal{G}$ is a conditional independence graph (CIG) where there is no edge between nodes i and j if and only if (iff) x_i and x_j are conditionally independent given the remaining p-2 variables. Gaussian graphical models (GGMs) are CIGs where $\boldsymbol{x}$ is multivariate Gaussian and $\{i, j\} \notin \mathcal{E}$ iff $[\boldsymbol{\Omega}]_{ij} = 0$ (where for zero-mean $\boldsymbol{x}$, precision matrix $\boldsymbol{\Omega} = (E\{\boldsymbol{x}\boldsymbol{x}^\top\})^{-1}$). There is extensive literature on this topic [2], [3]. Given n samples of $\boldsymbol{x}$, in high-dimensional settings (n is of the order of p), one estimates $\boldsymbol{\Omega}$ under some sparsity constraints, see, e.g., [3], [4], [5], and [6].

In differential network analysis one is interested in estimating the difference in two inverse covariance matrices [7], [8], [9]. Given observations $\boldsymbol{x}$ and $\boldsymbol{y}$ from two groups of subjects, one is interested in the difference $\boldsymbol{\Delta} = \boldsymbol{\Omega}_y - \boldsymbol{\Omega}_x$, where $\boldsymbol{\Omega}_x = (E\{\boldsymbol{x}\boldsymbol{x}^\top\})^{-1}$ and $\boldsymbol{\Omega}_y = (E\{\boldsymbol{y}\boldsymbol{y}^\top\})^{-1}$. The associated differential graph is $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$ where $\{i, j\} \in \mathcal{E}_\Delta$ iff $[\boldsymbol{\Delta}]_{ij} \neq 0$. In biostatistics, the differential network/graph describes the changes in conditional dependencies between components under different environmental or genetic conditions [8], [10], [11]. Given gene expression data or functional MRI signals, one is interested in the differences in the graphical models of healthy and impaired subjects, or models under different disease states [10], [11], [12].

In many applications, there may be more than one random variable associated with a node. This class of graphical models has been called multi-attribute graphical models in [13], [14], and [15] and vector graphs or networks in [16] and [17]. Consider p jointly Gaussian vectors $\boldsymbol{z}_i \in \mathbb{R}^m$, $i \in [p]$. We associate $\boldsymbol{z}_i$ with the ith node of graph $\mathcal{G} = (V, \mathcal{E})$, $V = [p]$, $\mathcal{E} \subseteq V \times V$. We now have m attributes per node. Now $\{i, j\} \in \mathcal{E}$ iff vectors $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are conditionally independent given the remaining p-2 vectors $\{\boldsymbol{z}_\ell, \ell \in V \backslash \{i, j\}\}$. Let $\boldsymbol{x} = [\boldsymbol{z}_1^\top \ \boldsymbol{z}_2^\top \ \cdots \ \boldsymbol{z}_p^\top]^\top \in \mathbb{R}^{mp}$. Let $\boldsymbol{\Omega} = (E\{\boldsymbol{x}\boldsymbol{x}^\top\})^{-1}$ and define the $m \times m$ subblock $\boldsymbol{\Omega}^{(ij)}$ of $\boldsymbol{\Omega}$ as

$$[\boldsymbol{\Omega}^{(ij)}]_{rs} = [\boldsymbol{\Omega}]_{(i-1)m+r,(j-1)m+s}, \quad r, s \in [m]. \tag{1}$$

The associate editor coordinating the review of this manuscript and approving it for publication was Kamran Siddique.

Then we have the following equivalence [13, Sec. 2.1]

$$\{i,j\} \notin \mathcal{E} \;\Leftrightarrow\; \mathbf{\Omega}^{(ij)} = \mathbf{0}\,. \tag{2}$$

Estimation of differential graphs from multi-attribute data was addressed in [15] in high-dimensional settings using a group-lasso penalty, and related approach of [18] also uses a group-lasso penalty (other past work has considered only single-attribute models). Given samples $\boldsymbol{x}(t)$, $t = 1, 2, \cdots, n_x$, of $\boldsymbol{x} = [z_1^\top \;\; z_2^\top \;\; \cdots \;\; z_p^\top]^\top \in \mathbb{R}^{mp}$ where $z_i \in \mathbb{R}^m$, $i \in [p]$, are jointly Gaussian, and similarly given samples $\boldsymbol{y}(t)$, $t = 1, 2, \cdots, n_y$, of $\boldsymbol{y} \in \mathbb{R}^{mp}$, the difference $\mathbf{\Delta} = \mathbf{\Omega}_y - \mathbf{\Omega}_x$ was estimated in [15] to determine the differential graph $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$ with edgeset $\mathcal{E}_\Delta = \{\{k,\ell\} : \|\mathbf{\Delta}^{(k\ell)}\|_F \neq 0\}$. It is well-known that use of non-convex penalties (unlike convex lasso penalty) can yield more accurate results, i.e., they can produce sparse set of solution like lasso, and approximately unbiased coefficients for large coefficients, unlike lasso [4], [19], [20]. The objective of this paper is to investigate use of non-convex (log-sum [20] and Smoothly Clipped Absolute Deviation (SCAD) [4], [19]) penalty functions for estimation of multi-attribute differential graphs.

A. RELATED WORK

Non-convex penalties have been extensively used for covariance and graph estimation (see [4], [21], [22], [23], [24] and references therein) and regression-related problems [25] (and references therein). However, only [26] and [27] have investigated use of non-convex penalties for differential graphs, but only for single-attribute differential graph estimation: SCAD and MCP (minimax concave penalty) in [26] and SCAD and log-sum penalty (LSP) in [27]. Counterparts to our Theorems 1 and 2 do not exist in [27]. For theoretical analysis, [26] does not require any irrepresentability condition, unlike this paper. On the other, we do not require a minimum amplitude condition on nonzero elements of $\mathbf{\Delta}$ but [26] does. Our numerical results show that our LSP-based differential graph estimator significantly outperforms both lasso and SCAD based methods. In [15] differential graphs were estimated using group-lasso (group ℓ_1) penalty. Here we extend [15] to non-convex penalties.

As discussed in [15, Sec. I-A], with the exception of [15], all prior work on high-dimensional differential graph estimation is focused on single-attribute models. One naive approach would be to estimate the two precision matrices separately by any existing estimator (see [3], [4], [21], [22], [23], [24] and references therein) and then calculate their difference to estimate the differential graph. (This approach is also applicable to multi-attribute graphs.) This approach estimates twice the number of parameters, hence needs larger sample sizes for same accuracy, and also imposes sparsity constraints on each precision matrix for the methods to work. The same comment applies to methods such as [12], where the two precision matrices and their differences are jointly estimated. If only the difference in the precision matrices is of interest, approaches exist where no sparsity constraints are imposed on individual precision matrices. For instance, direct estimation of the difference in the two precision matrices has been considered for single attribute graphs in [7], [8], [9], [10], [26], and [28], where only the difference is required to be sparse, not the two individual precision matrices. In [7], [8], [9], [26], and [28] precision difference matrix estimators are based on a D-trace loss [29], while [10] discusses a Dantzig selector type estimator. For more details, we refer the reader to [15, Sec. I-A].

We consider differential graph estimation without regard to the structure of the original graphs. For instance, there has been considerable interest in modeling the precision matrix as a Laplacian matrix [22], [30], [31] where non-convex penalties have been used. The off-diagonal elements of a Laplacian matrix are non-positive. The difference of two Laplacian matrices in not necessarily a Laplacian matrix because the off-diagonal elements of the difference can be positive, negative or zero. Therefore, the methods of [22], [30], and [31] do not apply to our problem.

B. OUR CONTRIBUTIONS

A penalized D-trace loss function approach for differential graph learning from multi-attribute data was presented in [15] using convex group-lasso penalty. In this paper we extend [15] to non-convex log-sum and SCAD penalties. Two proximal gradient descent methods are presented to optimize the objective function. Theoretical analysis establishing sufficient conditions for consistency in support recovery, convexity and estimation in high-dimensional settings is presented in Theorems 1 and 2. While the non-convex penalized D-trace loss function results in a non-convex optimization problem, Theorem 2 specifies conditions under which it becomes a convex optimization problem (see Remark 2 in Sec. IV). These conditions favor log-sum penalty over SCAD. Numerical results based on synthetic and real data are presented to illustrate the proposed approaches. In the synthetic data examples where the ground-truth is known, log-sum penalized D-trace loss significantly outperformed the lasso-penalized D-trace loss as well as SCAD penalized D-trace loss with F_1-score and Hamming distance as performance metrics.

C. OUTLINE AND NOTATION

The rest of the paper is organized as follows. A penalized D-trace loss function is presented in Sec. II for estimation of multi-attribute differential graph using non-convex penalties. Two proximal gradient descent methods are presented in Sec. III to optimize the non-convex objective function. In Sec. IV we analyze the properties of the estimator of the difference $\mathbf{\Delta} = \mathbf{\Omega}_y - \mathbf{\Omega}_x$, by following [15, Theorem 1] pertaining to the lasso penalty. Since the SCAD and log-sum penalties are non-convex, the objective function is non-convex and in general, any optimization of the objective function will yield only a stationary point. Theorem 1 analyzes the properties of such a stationary point under

some sufficient conditions, including an irrepresentability condition (similar condition also used in [5], [7], [13], [15], [28], and [29]). In Theorem 2 we investigate sufficient conditions under which the objective function is strictly convex, thereby ensuring that the stationary point of Theorem 1 is a unique minimum. Numerical results based on synthetic as well as real data are presented in Sec. V to illustrate the proposed approach. Proofs of Theorems 1 and 2 are given in Appendices A and B, respectively.

For a set V, $|V|$ or card(V) denotes its cardinality. Given $\boldsymbol{A} \in \mathbb{R}^{p\times p}$, we use $\phi_{\min}(\boldsymbol{A})$, $\phi_{\max}(\boldsymbol{A})$, $|\boldsymbol{A}|$ and tr$(\boldsymbol{A})$ to denote the minimum eigenvalue, maximum eigenvalue, determinant and trace of $\boldsymbol{A}$, respectively. For $\boldsymbol{B} \in \mathbb{R}^{p\times q}$, we define $\|\boldsymbol{B}\| = \sqrt{\phi_{\max}(\boldsymbol{B}^\top \boldsymbol{B})}$, $\|\boldsymbol{B}\|_F = \sqrt{\text{tr}(\boldsymbol{B}^\top \boldsymbol{B})}$, $\|\boldsymbol{B}\|_1 = \sum_{i,j} |B_{ij}|$, where B_{ij} is the (i,j)-th element of $\boldsymbol{B}$ (also denoted by $[\boldsymbol{B}]_{ij}$), $\|\boldsymbol{B}\|_\infty = \max_{i,j} |B_{ij}|$ and $\|\boldsymbol{B}\|_{1,\infty} = \max_i \sum_j |B_{ij}|$. The symbols $\otimes$ and $\boxtimes$ denote Kronecker product and Tracy-Singh product [32], respectively. In particular, given block partitioned matrices $\boldsymbol{A} = [\boldsymbol{A}_{ij}]$ and $\boldsymbol{B} = [\boldsymbol{B}_{k\ell}]$ with submatrices $\boldsymbol{A}_{ij}$ and $\boldsymbol{B}_{k\ell}$, Tracy-Singh product yields another block partitioned matrix $\boldsymbol{A}\boxtimes\boldsymbol{B} = [\boldsymbol{A}_{ij}\boxtimes\boldsymbol{B}]_{ij} = [[\boldsymbol{A}_{ij}\otimes\boldsymbol{B}_{k\ell}]_{k\ell}]_{ij}$ [33]. Given $\boldsymbol{A} = [\boldsymbol{A}_{ij}] \in \mathbb{R}^{mp\times mp}$ with $\boldsymbol{A}_{ij} \in \mathbb{R}^{m\times m}$, $vec(\boldsymbol{A}) \in \mathbb{R}^{m^2p^2}$ denotes the vectorization of $\boldsymbol{A}$ which stacks the columns of the matrix $\boldsymbol{A}$, and

$$bvec(\boldsymbol{A}) = [(vec(\boldsymbol{A}_{11}))^\top\ (vec(\boldsymbol{A}_{21}))^\top\ \cdots\ (vec(\boldsymbol{A}_{p1}))^\top \\ (vec(\boldsymbol{A}_{12}))^\top\ \cdots\ (vec(\boldsymbol{A}_{p2}))^\top\ \cdots\ (vec(\boldsymbol{A}_{pp}))^\top]^\top.$$

Let $S = \mathcal{E}_\Delta = \{\{k,\ell\} : \|\boldsymbol{\Delta}^{(k\ell)}\|_F \neq 0\}$ where $\boldsymbol{\Delta} = [\boldsymbol{\Delta}^{(k\ell)}] \in \mathbb{R}^{mp\times mp}$ with $\boldsymbol{\Delta}^{(k\ell)} \in \mathbb{R}^{m\times m}$ denoting the (k,l)th $m\times m$ submatrix of $\boldsymbol{\Delta}$. Then $\boldsymbol{\Delta}_S$ denotes the submatrix of $\boldsymbol{\Delta}$ with block rows and columns indexed by S, i.e., $\boldsymbol{\Delta}_S = [\boldsymbol{\Delta}^{(k\ell)}]_{(k,\ell)\in S}$. Suppose $\boldsymbol{\Gamma} = \boldsymbol{A}\boxtimes\boldsymbol{B}$ given block partitioned matrices $\boldsymbol{A} = [\boldsymbol{A}_{ij}]$ and $\boldsymbol{B} = [\boldsymbol{B}_{k\ell}]$. For any two subsets T_1 and T_2 of $V\times V$, $\boldsymbol{\Gamma}_{T_1,T_2}$ denotes the submatrix of $\boldsymbol{\Gamma}$ with block rows and columns indexed by T_1 and T_2, i.e., $\boldsymbol{\Gamma}_{T_1,T_2} = [\boldsymbol{A}_{j\ell}\otimes \boldsymbol{B}_{kq}]_{(j,k)\in T_1,(\ell,q)\in T_2}$. Following [13], an operator $\mathcal{C}(\cdot)$ is used in Sec. IV. Consider $\boldsymbol{A} \in \mathbb{R}^{mp\times mp}$ with (k,l)th $m\times m$ submatrix $\boldsymbol{A}^{(k\ell)}$. Then $\mathcal{C}(\cdot)$ operates on $\boldsymbol{A}$ as

$$\begin{bmatrix} \boldsymbol{A}^{(11)} & \cdots & \boldsymbol{A}^{(1p)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}^{(p1)} & \cdots & \boldsymbol{A}^{(pp)} \end{bmatrix} \xrightarrow{\mathcal{C}(\cdot)} \begin{bmatrix} \|\boldsymbol{A}^{(11)}\|_F & \cdots & \|\boldsymbol{A}^{(1p)}\|_F \\ \vdots & \ddots & \vdots \\ \|\boldsymbol{A}^{(p1)}\|_F & \cdots & \|\boldsymbol{A}^{(pp)}\|_F \end{bmatrix}$$

with $\mathcal{C}(\boldsymbol{A}^{(k\ell)}) = \|\boldsymbol{A}^{(k\ell)}\|_F$ and $\mathcal{C}(\boldsymbol{A}) \in \mathbb{R}^{p\times p}$. Now consider $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{R}^{mp\times mp}$ with (k,l)th $m\times m$ submatrices $\boldsymbol{A}_1^{(k\ell)}$ and $\boldsymbol{A}_2^{(k\ell)}$, respectively, and Tracy-Singh product $\boldsymbol{A}_1\boxtimes\boldsymbol{A}_2 \in \mathbb{R}^{(mp)^2\times(mp)^2}$. Then $\mathcal{C}(\cdot)$ operates on $\boldsymbol{A}_1\boxtimes\boldsymbol{A}_2$ as $\mathcal{C}(\boldsymbol{A}_1\boxtimes\boldsymbol{A}_2) \in \mathbb{R}^{p^2\times p^2}$ with $\mathcal{C}(\boldsymbol{A}_1^{(k_1\ell_1)}\otimes\boldsymbol{A}_2^{(k_2\ell_2)}) = \|\boldsymbol{A}_1^{(k_1\ell_1)}\otimes\boldsymbol{A}_2^{(k_2\ell_2)}\|_F$ $(=\|\boldsymbol{A}_1^{(k_1\ell_1)}\|_F\,\|\boldsymbol{A}_2^{(k_2\ell_2)}\|_F)$. That is, each $m^2\times m^2$ submatrix $\boldsymbol{A}_1^{(k_1\ell_1)}\otimes\boldsymbol{A}_2^{(k_2\ell_2)}$ of $\boldsymbol{A}_1\boxtimes\boldsymbol{A}_2$ is mapped into its Frobenius norm.

II. PENALIZED D-TRACE LOSS

Let $\boldsymbol{x} = [\boldsymbol{z}_1^\top\ \boldsymbol{z}_2^\top\ \cdots\ \boldsymbol{z}_p^\top]^\top \in \mathbb{R}^{mp}$ where $\boldsymbol{z}_i \in \mathbb{R}^m$, $i \in [p]$, are zero-mean, jointly Gaussian. Given i.i.d. samples $\boldsymbol{x}(t)$, $t = 1,2,\cdots,n_x$, of $\boldsymbol{x}$, and similarly given i.i.d. samples $\boldsymbol{y}(t)$, $t = 1,2,\cdots,n_y$, of $\boldsymbol{y} \in \mathbb{R}^{mp}$, form the sample covariance estimates

$$\hat{\boldsymbol{\Sigma}}_x = \frac{1}{n_x}\sum_{t=1}^{n_x} \boldsymbol{x}(t)\boldsymbol{x}^\top(t), \quad \hat{\boldsymbol{\Sigma}}_y = \frac{1}{n_y}\sum_{t=1}^{n_y} \boldsymbol{y}(t)\boldsymbol{y}^\top(t). \tag{3}$$

and denote their true values as $\boldsymbol{\Sigma}_x^* = \boldsymbol{\Omega}_x^{-*}(= (\boldsymbol{\Omega}_x^*)^{-1})$ and $\boldsymbol{\Sigma}_y^* = \boldsymbol{\Omega}_y^{-*}$. We wish to estimate $\boldsymbol{\Delta} = \boldsymbol{\Omega}_y^* - \boldsymbol{\Omega}_x^*$ and graph $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$, based on $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$. Following [15] (see also [7]and [28, Sec. 2.1]), we use the convex D-trace (difference-in-trace) loss function

$$L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) = \frac{1}{2}\text{tr}(\hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y\boldsymbol{\Delta}^\top) - \text{tr}(\boldsymbol{\Delta}(\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y)). \tag{4}$$

The function $L(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^*, \boldsymbol{\Sigma}_y^*)$ is strictly convex in $\boldsymbol{\Delta}$ and has a unique minimum at $\boldsymbol{\Delta}^* = \boldsymbol{\Omega}_y^* - \boldsymbol{\Omega}_x^*$ [7], [28]. When we use sample covariances, we estimate $\boldsymbol{\Delta}$ by minimizing the penalized loss function

$$L_\lambda(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) = L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) + \sum_{k,\ell=1}^{p} \rho_\lambda\big(\|\boldsymbol{\Delta}^{(k\ell)}\|_F\big) \tag{5}$$

where, for $u \in \mathbb{R}$, $\rho_\lambda(u)$ is a penalty function that is function of $|u|$, $\lambda > 0$ is a tuning parameter, and $\|\boldsymbol{\Delta}^{(k\ell)}\|_F$ promotes blockwise sparsity in $\boldsymbol{\Delta}$ [34] where, if we partition $\boldsymbol{\Delta}$ into $m\times m$ submatrices, $\boldsymbol{\Delta}^{(k\ell)}$ denotes its (k,ℓ)th submatrix, associated with edge $\{k,\ell\}$ of edgeset $\mathcal{E}_\Delta$ of the differential graph $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$. **For ease of notation, henceforth, we also use $L_\lambda(\boldsymbol{\Delta})$, $L(\boldsymbol{\Delta})$ and $L^*(\boldsymbol{\Delta})$ for $L_\lambda(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)$, $L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)$ and $L(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^*, \boldsymbol{\Sigma}_y^*)$, respectively.**

The following penalty functions are considered:

(i) *Lasso*. For some $\lambda > 0$, $\rho_\lambda(u) = \lambda|u|$, $u \in \mathbb{R}$. It is the well-known ℓ_1-penalty resulting in a convex function that is widely used (also used in [15]).

(ii) *Log-sum*. For some $\lambda > 0$ and $1 \gg \epsilon > 0$, $\rho_\lambda(u) = \lambda\epsilon\ \ln\big(1 + \frac{|u|}{\epsilon}\big)$. It is a non-convex function.

(iii) *Smoothly Clipped Absolute Deviation (SCAD)*. For some $\lambda > 0$ and $a > 2$, $\rho_\lambda(u) = \lambda|u|$ for $|u| \leq \lambda$, $= (2a\lambda|u| - |u|^2 - \lambda^2)/(2(a-1))$ for $\lambda < |u| < a\lambda$, and $= \lambda^2(a+1)/2$ for $|u| \geq a$. It is a non-convex function.

In the terminology of [35], all of the above three penalties are "μ-amenable" for some $\mu \geq 0$. As defined in [35, Sec. 2.2], $\rho_\lambda(u)$ is μ-amenable for some $\mu \geq 0$ if the following properties hold:

(i) The function $\rho_\lambda(u)$ is symmetric around zero, i.e., $\rho_\lambda(u) = \rho_\lambda(-u)$ and $\rho_\lambda(0) = 0$.

(ii) The function $\rho_\lambda(u)$ is non-decreasing on $\mathbb{R}_+$.

(iii) The function $\rho_\lambda(u)/u$ is non-increasing on $\mathbb{R}_+$.

(iv) The function $\rho_\lambda(u)$ is differentiable for $u \neq 0$.

(v) The function $\rho_\lambda(u) + \frac{\mu}{2}u^2$ is convex, for some $\mu \geq 0$.

(iv) $\lim_{u\to 0^+} \rho'(u) = \lambda$ where $\rho'(u) := \frac{d\rho_\lambda(u)}{du}$.

It is shown in [35, Appendix A.1], that all of the above three penalties are μ-amenable with $\mu = 0$ for lasso and $\mu = 1/(a-1)$ for SCAD. In [35] the log-sum penalty is defined as $\rho_\lambda(u) = \ln(1 + \lambda|u|)$ whereas in [20], it is defined as $\rho_\lambda(u) = \lambda\ \ln\left(1 + \frac{|u|}{\epsilon}\right)$. We follow [20] but modify it so

that property (vi) in the definition of μ-amenable penalties holds. In our case $\mu = \frac{\lambda}{\epsilon}$ for the log-sum penalty.

Suppose

$$\hat{\mathbf{\Delta}} = \arg\min_{\mathbf{\Delta}} L_\lambda(\mathbf{\Delta}). \tag{6}$$

Even though $\mathbf{\Delta}$ is symmetric, $\hat{\mathbf{\Delta}}$ may not be. We can symmetrize it by setting $\hat{\mathbf{\Delta}}_{sym} = \frac{1}{2}(\hat{\mathbf{\Delta}} + \hat{\mathbf{\Delta}}^\top)$, after obtaining $\hat{\mathbf{\Delta}}$.

III. OPTIMIZATION

The objective function $L_\lambda(\mathbf{\Delta})$ is non-convex for the non-convex SCAD and log-sum penalties. In this section we discuss two possible optimization approaches to attain a local minimum (actually a stationary point) of $L_\lambda(\mathbf{\Delta})$. Both are based on a proximal gradient descent (PGD) approach which was found to perform better than an alternating direction method of multipliers (ADMM) approach in simulation examples in [15] (with F_1-score as the performance metric).

A. LOCAL LINEAR APPROXIMATION (LLA)

Here, for non-convex $\rho_\lambda(u)$, we use a local linear approximation (LLA) to $\rho_\lambda(u)$ as in [4] and [21], to yield

$$\rho_\lambda(u) \approx \rho_\lambda(|u_0|) + \rho'_\lambda(|u_0|)(|u| - |u_0|), \tag{7}$$

where u_0 is an initial guess, and the gradient of the penalty function is

$$\rho'_\lambda(|u_0|) = \begin{cases} \frac{\lambda\epsilon}{|u_0|+\epsilon} \text{ for log-sum,} \\ \begin{cases} \lambda, & \text{if } |u_0| \le \lambda \\ \frac{a\lambda - |u_0|}{a-1}, & \text{if } \lambda < |u_0| \le a\lambda \\ 0, & \text{if } a\lambda < |u_0| \end{cases} \\ \text{for SCAD.} \end{cases} \tag{8}$$

Therefore, with u_0 fixed, we need to consider only the term dependent upon u for optimization w.r.t. u:

$$\rho_\lambda(u) \Rightarrow \rho'_\lambda(|u_0|)\,|u|. \tag{9}$$

By [21, Theorem 1], the LLA provides a majorization of the non-convex penalty, thereby yielding a majorization-minimization approach.

Thus in LSP, with some initial guess $\bar{\mathbf{\Delta}}$, we replace

$$\rho_\lambda(\|\mathbf{\Delta}^{(k\ell)}\|_F) \to \lambda_{k\ell} := \frac{\lambda\epsilon}{\|\bar{\mathbf{\Delta}}^{(k\ell)}\|_F + \epsilon}. \tag{10}$$

The solution $\hat{\mathbf{\Delta}}_{lasso}$ to the convex lasso-penalized objective function may be used as an initial guess with $\bar{\mathbf{\Delta}} = \hat{\mathbf{\Delta}}_{lasso}$. Similarly, for SCAD, we have

$$\lambda_{k\ell} = \begin{cases} \lambda, & \text{if } \|\bar{\mathbf{\Delta}}^{(k\ell)}\|_F \le \lambda \\ \frac{a\lambda - \|\bar{\mathbf{\Delta}}^{(k\ell)}\|_F}{a-1}, & \text{if } \lambda < \|\bar{\mathbf{\Delta}}^{(k\ell)}\|_F \le a\lambda \\ 0, & \text{if } a\lambda < \|\bar{\mathbf{\Delta}}^{(k\ell)}\|_F \end{cases}. \tag{11}$$

With LLA, the objective function is transformed to

$$\bar{L}_\lambda(\mathbf{\Delta}) = L(\mathbf{\Delta}) + \sum_{k,\ell=1}^{p} \lambda_{k\ell}\,\|\mathbf{\Delta}^{(k\ell)}\|_F. \tag{12}$$

Algorithm 1 PGD Algorithm Under LLA

Input: $\hat{\mathbf{\Sigma}}_x$, $\hat{\mathbf{\Sigma}}_y$, tolerance δ, maximum number of iterations i_{max}, lasso estimate $\hat{\mathbf{\Delta}}_{lasso}$ for SCAD and log-sum penalties
Output: Estimated $\hat{\mathbf{\Delta}}_{sym}$ and $\hat{\mathcal{E}}_\Delta$.

1: Set $\eta = 1/L_{c1}$ (L_{c1} as in (17)), $\mathbf{\Delta}^{(0)} = \mathbf{0}$ for lasso, $= \hat{\mathbf{\Delta}}_{lasso}$ for SCAD and log-sum penalties.
2: converged = FALSE, $i = 0$
3: **while** converged = FALSE AND $i \le i_{max}$, **do**
4: Set $\boldsymbol{A}_1 = \mathbf{\Delta}^{(i)} - \eta\nabla L(\mathbf{\Delta}^{(i)})$, with $\nabla L(\mathbf{\Delta})$ as in (16).
5: Update $(\mathbf{\Delta}^{(i+1)})^{(k\ell)} = \left(1 - \frac{\lambda_{k\ell}\eta}{\|\boldsymbol{A}_1^{(k\ell)}\|_F}\right)_+ \boldsymbol{A}_1^{(k\ell)}$ for $k, \ell \in [p]$.
6: If $\frac{\bar{L}(\mathbf{\Delta}^{(i+1)}) - \bar{L}(\mathbf{\Delta}^{(i)})}{\bar{L}(\mathbf{\Delta}^{(i)})} \le \delta$, set converged = TRUE.
7: $i \leftarrow i + 1$
8: **end while**
9: Set $\hat{\mathbf{\Delta}}_{sym} = \frac{1}{2}(\mathbf{\Delta} + \mathbf{\Delta}^\top)$. If $\|\hat{\mathbf{\Delta}}_{sym}^{(jk)}\|_F > 0$, assign edge $\{j,k\} \in \hat{\mathcal{E}}_\Delta$, else $\{j,k\} \notin \hat{\mathcal{E}}_\Delta$.

We will use an iterative PGD approach [36], [37] to minimize $\bar{L}_\lambda(\mathbf{\Delta})$. It is a first-order method that is based on objective function values and gradient evaluations. It has been used for differential graph estimation with lasso penalty in [8], [15], and [18] where $\lambda_{k\ell} = \lambda$ for all edges $\{k, \ell\}$.

In the PGD method to minimize $\bar{L}_\lambda(\mathbf{\Delta})$, given the old estimate $\mathbf{\Delta}^{old}$, in the next iteration, the new estimate $\mathbf{\Delta}^{new}$ is given by

$$\mathbf{\Delta}^{new} = \arg\min_{\mathbf{\Delta}} \left(\frac{1}{2}\|\mathbf{\Delta} - \boldsymbol{A}_1\|_F^2 + \eta \sum_{k,\ell=1}^{p} \lambda_{k\ell}\|\mathbf{\Delta}^{(k\ell)}\|_F\right) \tag{13}$$

where

$$\boldsymbol{A}_1 = \mathbf{\Delta}^{old} - \eta\nabla L(\mathbf{\Delta}^{old}), \tag{14}$$

for a step-size of η and $\nabla L(\mathbf{\Delta}^{old})$ is the gradient of $L(\mathbf{\Delta})$ at the old value. The solution is given by [8], [18], and [34]

$$(\mathbf{\Delta}^{(k\ell)})^{new} = \left(1 - \frac{\lambda_{k\ell}\eta}{\|\boldsymbol{A}_1^{(k\ell)}\|_F}\right)_+ \boldsymbol{A}_1^{(k\ell)}, \quad k, \ell \in [p], \tag{15}$$

where $b_+ = \max(0, b)$, $b \in \mathbb{R}$. We have [8], [18], [34]

$$\nabla L(\mathbf{\Delta}) = \hat{\mathbf{\Sigma}}_x\mathbf{\Delta}\hat{\mathbf{\Sigma}}_y - (\hat{\mathbf{\Sigma}}_x - \hat{\mathbf{\Sigma}}_y). \tag{16}$$

The function $L(\mathbf{\Delta})$ is Lipschitz-continuous with Lipschitz constant L_c given by [8] and [38]

$$L_{c1} = \phi_{max}(\hat{\mathbf{\Sigma}}_x)\,\phi_{max}(\hat{\mathbf{\Sigma}}_y). \tag{17}$$

Therefore, a fixed step-size choice $0 < \eta \le L_{c1}^{-1}$ guarantees convergence of the PGD method to a (local) minimum [36], which is a global minimum of the LLA cost since $\bar{L}_\lambda(\mathbf{\Delta})$ is convex.

A pseudocode for the PGD algorithm is given in Algorithm 1. We minimize $\bar{L}_\lambda(\mathbf{\Delta})$ w.r.t. $\mathbf{\Delta}$ using Algorithm 1 as follows:

(i) Lasso: $\lambda_{k\ell} = \lambda\ \forall k, \ell$.

(ii) Log-sum: First run lasso to convergence, then use LLA with $\bar{\mathbf{\Delta}} = \hat{\mathbf{\Delta}}_{lasso}$ to obtain $\lambda_{k\ell}$s, and repeat Algorithm 1.
(iii) SCAD: Follow the procedure for log-sum but with $\lambda_{k\ell}$s computed for SCAD.

B. NONCONVEXITY REDISTRIBUTION

Let $\tilde{\rho}_\lambda(u) = \rho_\lambda(u) - \lambda|u|$. Then $\tilde{\rho}_\lambda(u)$ is everywhere differentiable with $\tilde{\rho}'_\lambda(0) = 0$. Similar to [35] and [38], we rewrite (5) as

$$L_\lambda(\mathbf{\Delta}) = \tilde{L}_\lambda(\mathbf{\Delta}) + \lambda \sum_{k,\ell=1}^{p} \|\mathbf{\Delta}^{(k\ell)}\|_F \tag{18}$$

where

$$\tilde{L}_\lambda(\mathbf{\Delta}) = L(\mathbf{\Delta}) + \sum_{k,\ell=1}^{p} \tilde{\rho}_\lambda\big(\|\mathbf{\Delta}^{(k\ell)}\|_F\big), \tag{19}$$

$\tilde{L}_\lambda(\mathbf{\Delta})$ is non-convex but smooth and $\sum_{k,\ell=1}^{p} \|\mathbf{\Delta}^{(k\ell)}\|_F$ is convex but nonsmooth [35]. In (5), $L(\mathbf{\Delta})$ is convex and smooth but $\rho_\lambda\big(\|\mathbf{\Delta}^{(k\ell)}\|_F\big)$ is non-convex and nonsmooth. Now in the PGD approach, given the old estimate $\mathbf{\Delta}^{old}$, in the next iteration, the new estimate $\mathbf{\Delta}^{new}$ is given by

$$\mathbf{\Delta}^{new} = \arg\min_{\mathbf{\Delta}} \Big(\frac{1}{2}\|\mathbf{\Delta} - \boldsymbol{A}_2\|_F^2 + \eta\lambda \sum_{k,\ell=1}^{p} \|\mathbf{\Delta}^{(k\ell)}\|_F\Big) \tag{20}$$

where

$$\boldsymbol{A}_2 = \mathbf{\Delta}^{old} - \eta \nabla \tilde{L}_\lambda(\mathbf{\Delta}^{old}), \tag{21}$$

for a step-size of η and $\nabla \tilde{L}_\lambda(\mathbf{\Delta}^{old})$ is the gradient of $\tilde{L}_\lambda(\mathbf{\Delta})$ at the old value. The solution is given by [8], [18], and [34]

$$(\mathbf{\Delta}^{(k\ell)})^{new} = \Big(1 - \frac{\lambda\eta}{\|\boldsymbol{A}_2^{(k\ell)}\|_F}\Big)_+ \boldsymbol{A}_2^{(k\ell)}, \quad k, \ell \in [p]\,. \tag{22}$$

We have

$$\nabla \tilde{L}_\lambda(\mathbf{\Delta}) = \hat{\mathbf{\Sigma}}_x \mathbf{\Delta} \hat{\mathbf{\Sigma}}_y - (\hat{\mathbf{\Sigma}}_x - \hat{\mathbf{\Sigma}}_y) + \boldsymbol{G}, \tag{23}$$

where

$$\boldsymbol{G}^{(k\ell)} = \begin{cases} \mathbf{0} \text{ for lasso} \\ \lambda\big(\frac{\epsilon}{\epsilon + \|\mathbf{\Delta}^{(k\ell)}\|_F} - 1\big)\frac{\mathbf{\Delta}^{(k\ell)}}{\|\mathbf{\Delta}^{(k\ell)}\|_F} \text{ for log-sum} \\ \begin{cases} \mathbf{0}, & \text{if } \|\mathbf{\Delta}^{(k\ell)}\|_F \le \lambda \\ \boldsymbol{B}^{(k\ell)}, & \text{if } \lambda < \|\mathbf{\Delta}^{(k\ell)}\|_F \le a\lambda \\ -\lambda\frac{\mathbf{\Delta}^{(k\ell)}}{\|\mathbf{\Delta}^{(k\ell)}\|_F}, & \text{if } a\lambda < \|\mathbf{\Delta}^{(k\ell)}\|_F \end{cases} \\ \text{for SCAD} \end{cases} \tag{24}$$

and

$$\boldsymbol{B}^{(k\ell)} = \Big(\frac{a\lambda - \|\mathbf{\Delta}^{(k\ell)}\|_F}{a-1} - \lambda\Big)\frac{\mathbf{\Delta}^{(k\ell)}}{\|\mathbf{\Delta}^{(k\ell)}\|_F}\,. \tag{25}$$

Algorithm 2 PGD Algorithm After Non-Convexity Redistribution

Input: $\hat{\mathbf{\Sigma}}_x$, $\hat{\mathbf{\Sigma}}_y$, tolerance δ, maximum number of iterations i_{max}, lasso estimate $\hat{\mathbf{\Delta}}_{lasso}$ for SCAD and log-sum penalties
Output: Estimated $\hat{\mathbf{\Delta}}_{sym}$ and $\hat{\mathcal{E}}_\Delta$.

1: Set $\eta = 1/L_{c2}$ (L_{c2} as in (26)), $\mathbf{\Delta}^{(0)} = \mathbf{0}$ for lasso, $= \hat{\mathbf{\Delta}}_{lasso}$ for SCAD and log-sum penalties.
2: converged = FALSE, $i = 0$
3: **while** converged = FALSE AND $i \le i_{max}$, **do**
4: Set $\boldsymbol{A}_2 = \mathbf{\Delta}^{(i)} - \eta\nabla\tilde{L}_\lambda(\mathbf{\Delta}^{(i)})$, with $\nabla\tilde{L}_\lambda(\mathbf{\Delta})$ as in (23).
5: Update $(\mathbf{\Delta}^{(i+1)})^{(k\ell)} = \big(1 - \frac{\lambda\eta}{\|\boldsymbol{A}_2^{(k\ell)}\|_F}\big)_+ \boldsymbol{A}_2^{(k\ell)}$ for $k, \ell \in [p]$.
6: If $\frac{L_\lambda(\mathbf{\Delta}^{(i+1)}) - L_\lambda(\mathbf{\Delta}^{(i)})}{L_\lambda(\mathbf{\Delta}^{(i)})} \le \delta$, set converged = TRUE.
7: $i \leftarrow i + 1$
8: **end while**
9: Set $\hat{\mathbf{\Delta}}_{sym} = \frac{1}{2}(\mathbf{\Delta} + \mathbf{\Delta}^\top)$. If $\|\hat{\mathbf{\Delta}}_{sym}^{(jk)}\|_F > 0$, assign edge $\{j,k\} \in \hat{\mathcal{E}}_\Delta$, else $\{j,k\} \notin \hat{\mathcal{E}}_\Delta$.

The function $\tilde{L}_\lambda(\mathbf{\Delta})$ is Lipschitz-continuous with Lipschitz constant L_{c2} given by [8] and [38]

$$L_{c2} = \begin{cases} \phi_{max}(\hat{\mathbf{\Sigma}}_x)\,\phi_{max}(\hat{\mathbf{\Sigma}}_y) & : \text{Lasso} \\ \phi_{max}(\hat{\mathbf{\Sigma}}_x)\,\phi_{max}(\hat{\mathbf{\Sigma}}_y) + \dfrac{2m\lambda}{\epsilon} & : \text{Log-sum} \\ \phi_{max}(\hat{\mathbf{\Sigma}}_x)\,\phi_{max}(\hat{\mathbf{\Sigma}}_y) + \dfrac{2m}{a-1} & : \text{SCAD} \end{cases} \tag{26}$$

where we have used [38, Cor. 2] for the log-sum and SCAD penalties. Therefore, a fixed step-size choice $0 < \eta \le L_{c2}^{-1}$ guarantees convergence of the PGD method to a local minimum [39] provided $L_\lambda(\mathbf{\Delta})$ is bounded below and coercive. We show in Sec. IV that for large n ($= \min(n_x, n_y)$) this is true with high probability (w.h.p.). A pseudocode for this PGD algorithm is given in Algorithm 2. Since the problem is convex for the lasso penalty, we choose to initialize with the lasso result for log-sum and SCAD.

C. COMPUTATIONAL COMPLEXITY, CONVERGENCE AND MODEL SELECTION

The computational complexity of the PGD method has been discussed in [8] for single-attribute differential graphs, and it is of the same order for MA graphs, because the difference lies only in element-wise penalty versus group penalty. Noting that we have $mp \times mp$ precision matrices, by [8], the computational complexity of the PGD methods of [8] and [18] is either $\mathcal{O}((mp)^3)$ when as implemented in Algorithms 1 and 2, or $\mathcal{O}((n_x+n_y)(mp)^2)$ when an alternative implementation of the cost gradient in (16) is used (see [8, Sec. 2.2]). For $n_x + n_y \ge mp$, there is no advantage to this alternative approach.

In the LLA approach, each approximation yields a convex objective function, therefore, convergence to a global minimum is guaranteed. Overall it is a majorization-minimization approach, hence, after repeated LLA's, one

gets a local minimum of the original non-convex objective function. In practice, two iterations seem to be enough: first run Algorithm 1 for lasso, then using lasso-based LLA, run Algorithm 1 once more. Convergence of the PGD method of Algorithm 2 to a local minimum is guaranteed [39] provided $L_\lambda(\mathbf{\Delta})$ is bounded below and coercive. We show in Sec. IV that for large n this is true w.h.p.

For model selection we follow the BIC-like criterion as given in [15, Sec. III-E] (which follows [7] who invokes [10]):

$$BIC(\lambda) = (n_x + n_y)\,\|\hat{\mathbf{\Sigma}}_x\hat{\mathbf{\Delta}}\hat{\mathbf{\Sigma}}_y - (\hat{\mathbf{\Sigma}}_x - \hat{\mathbf{\Sigma}}_y)\|_F + \ln(n_x + n_y)\,|\hat{\mathbf{\Delta}}|_0 \tag{27}$$

where $|\boldsymbol{A}|_0$ denotes number of nonzero elements in $\boldsymbol{A}$ and $\hat{\mathbf{\Delta}}$ obeys (6). Choose λ to minimize $BIC(\lambda)$. Since (27) is not scale invariant, we scale both $\hat{\mathbf{\Sigma}}_x$ and $\hat{\mathbf{\Sigma}}_y$ (and $\hat{\mathbf{\Delta}}$ commensurately) by $\bar{\mathbf{\Sigma}}^{-1}$ where $\bar{\mathbf{\Sigma}} = \text{diag}\{\hat{\mathbf{\Sigma}}_x\}$ is a diagonal matrix of diagonal elements of $\hat{\mathbf{\Sigma}}_x$.

In our simulations we search over $\lambda \in [\lambda_\ell, \lambda_u]$, where λ_ℓ and λ_u are selected via a heuristic as in [14]. Find the smallest λ, labeled λ_{sm} for which we get a no-edge model; then we set $\lambda_u = \lambda_{sm}/2$ and $\lambda_\ell = \lambda_u/10$. For real data results we picked $\lambda_u = \lambda_{sm}$ and $\lambda_\ell = \lambda_u/5$.

For the numerical results presented later, we picked $i_{\max} = 200$ and $\delta = 10^{-3}$ in Algorithms 1 and 2. For the SCAD penalty $a = 3.7$ (as in [4]) and for log-sum penalty $\epsilon = 0.001$.

IV. THEORETICAL ANALYSIS

Here we analyze the properties of $\hat{\mathbf{\Delta}}$ specified in (6), by following [15, Theorem 1] pertaining to the lasso penalty. Since the SCAD and log-sum penalties are non-convex, the objective function is non-convex and in general, any optimization of the objective function will yield only a stationary point. Theorem 1 analyzes the properties of such a stationary point under some sufficient conditions, including an irrepresentability condition (similar condition also used in [5], [7], [13], [15], [28], [29]). In Theorem 2 we investigate sufficient conditions under which the objective function is strictly convex, thereby ensuring that the stationary point of Theorem 1 is a unique minimum.

Define the true differential edgeset ($\mathbf{\Delta}^*$ denotes the true value of $\mathbf{\Delta}$)

$$S = \mathcal{E}_{\Delta^*} = \{\{k,\ell\} : \|\mathbf{\Delta}^{*(k\ell)}\|_F \neq 0\}, \quad s = |S|\,. \tag{28}$$

Define $n = \min(n_x, n_y)$, and

$$\mathbf{\Gamma}^* = \mathbf{\Sigma}_y^* \boxtimes \mathbf{\Sigma}_x^*, \quad \hat{\mathbf{\Gamma}} = \hat{\mathbf{\Sigma}}_y \boxtimes \hat{\mathbf{\Sigma}}_x\,. \tag{29}$$

Also, recall the operator $\mathcal{C}(\cdot)$ defined in Sec. I-C. In the rest of this section, we allow p, s and λ to be a functions of sample size n, denoted as p_n, s_n and λ_n, respectively. Define

$$M = \max\{\|\mathcal{C}(\mathbf{\Sigma}_x^*)\|_\infty, \|\mathcal{C}(\mathbf{\Sigma}_y^*)\|_\infty\}, \tag{30}$$

$$M_\Sigma = \max\{\|\mathcal{C}(\mathbf{\Sigma}_x^*)\|_{1,\infty}, \|\mathcal{C}(\mathbf{\Sigma}_y^*)\|_{1,\infty}\}, \tag{31}$$

$$\kappa_\Gamma = \|\mathcal{C}((\Gamma_{S,S}^*)^{-1})\|_{1,\infty}, \tag{32}$$

$$\alpha = 1 - \max_{e\in S^c} \|\mathcal{C}(\mathbf{\Gamma}_{e,S}^*(\mathbf{\Gamma}_{S,S}^*)^{-1})\|_1, \tag{33}$$

$$\bar{\sigma}_{xy} = \max\{\max_i [\mathbf{\Sigma}_x^*]_{ii},\ \max_i [\mathbf{\Sigma}_y^*]_{ii}\}, \tag{34}$$

$$C_0 = 40\,m\,\bar{\sigma}_{xy}\sqrt{2\big(\tau + \ln(4m^2)/\ln(p_n)\big)} \tag{35}$$

where S and $\mathbf{\Gamma}^*$ have been defined in (28) and (29). In (33), we require $0 < \alpha < 1$, and the expression

$$\max_{e\in S^c} \|\mathcal{C}(\mathbf{\Gamma}_{e,S}^*(\mathbf{\Gamma}_{S,S}^*)^{-1})\|_1 \leq 1 - \alpha$$

for some $\alpha \in (0,1)$ is called the *irrepresentability condition*. Similar conditions are also used in [5], [7], [13], [28], and [29].

Let $\partial L_\lambda(\mathbf{\Delta})$ denote the sub-differential of $L_\lambda(\mathbf{\Delta})$. Recall that we write $\tilde{\rho}_\lambda(u) = \rho_\lambda(u) - \lambda|u|$ so that $\tilde{\rho}_\lambda(u)$ is everywhere differentiable with $\tilde{\rho}'_\lambda(0) = 0$. Suppose that $\hat{\mathbf{\Delta}}$ is a solution to

$$\mathbf{0} \in \partial L_\lambda(\mathbf{\Delta}) = \frac{\partial L(\mathbf{\Delta})}{\partial \mathbf{\Delta}} + \frac{\partial}{\partial \mathbf{\Delta}}\Big(\sum_{k,\ell=1}^{p} \tilde{\rho}_\lambda\big(\|\mathbf{\Delta}^{(k\ell)}\|_F\big)\Big) + \partial\Big(\lambda \sum_{k,\ell=1}^{p} \|\mathbf{\Delta}^{(k\ell)}\|_F\Big), \tag{36}$$

which is a first-order necessary condition for a stationary point of $L_\lambda(\mathbf{\Delta})$. Theorem 1 addresses some properties of this $\hat{\mathbf{\Delta}}$.

Theorem 1: Suppose (36) is satisfied for $\mathbf{\Delta} = \hat{\mathbf{\Delta}}$. For the system model of Sec. II, under (28) and the irrepresentability condition (33) for some $\alpha \in (0,1)$, if

$$\lambda_n = \max\Big\{\frac{8}{\alpha}, \frac{3}{\alpha\bar{C}_\alpha}\, s_n\kappa_\Gamma M C_{M\kappa}\Big\} C_0\sqrt{\frac{\ln(p_n)}{n}} \tag{37}$$

$$n = \min(n_x, n_y) > \max\Big\{\frac{1}{\min\{M^2, 1\}}, 81\,M^2 s_n^2\kappa_\Gamma^2, \frac{9s_n^2}{(\alpha\bar{C}_\alpha)^2}(\kappa_\Gamma M C_{M\kappa})^2\Big\} C_0^2\ln(p_n) \tag{38}$$

where $\bar{C}_\alpha = \frac{1-\alpha}{2(2M+1)-2\alpha M}$ and $C_{M\kappa} = 1.5\big(1 + \kappa_\Gamma \min\{s_n M^2, M_\Sigma^2\}\big)$, then $\hat{\mathbf{\Delta}}$ is such that with probability $> 1 - 2/p_n^{\tau-2}$, for any $\tau > 2$, we have

(i) $\|\mathcal{C}(\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*)\|_\infty \leq (C_{b1} + C_{b2})C_0\sqrt{\frac{\ln(p_n)}{n}}$

where $C_{b1} = 3\kappa_\Gamma \max\Big\{\frac{8}{\alpha}, \frac{3}{\alpha\bar{C}_\alpha}\, s_n\kappa_\Gamma M C_{M\kappa}\Big\}$,

$C_{b2} = 9\, s_n\kappa_\Gamma^2 M^2\,.$

(ii) $\hat{\mathbf{\Delta}}_{S^c} = \mathbf{0}$.

(iii) $\|\mathcal{C}(\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*)\|_F \leq \sqrt{s_n}\,\|\mathcal{C}(\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*)\|_\infty$.

(iv) Additionally, if $\min_{(k,\ell)\in S}\|(\mathbf{\Delta}^*)^{(k\ell)}\|_F \geq 2(C_{b1} + C_{b2})C_0\sqrt{\frac{\ln(p_n)}{n}}$, then $P(\mathcal{G}_{\hat{\Delta}} = \mathcal{G}_{\Delta^*}) > 1 - 2/p_n^{\tau-2}$ (support recovery) •

The proof of Theorem 1 is given in Appendix A.

Remark 1 (Convergence Rate): As discussed in [15], if M, M_Σ and κ_Γ stay bounded with increasing sample size n, we have $\|\mathcal{C}(\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*)\|_F = \mathcal{O}_P(s_n^{1.5}\sqrt{\ln(p_n)/n})$. Therefore, for $\|\mathcal{C}(\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*)\|_F \to 0$ as $n \to \infty$, we must have $s_n^{1.5}\sqrt{\ln(p_n)/n} \to 0$. Notice that M_Σ constraints covariances $\mathbf{\Sigma}_x^*$ and $\mathbf{\Sigma}_y^*$ which can be dense even if $\mathbf{\Omega}_x^*$ and $\mathbf{\Omega}_y^*$ are

sparse (they need not be sparse for differential estimation), making them possibly unbounded with increasing sample size n. In this case we use $\min\{s_n M^2, M_\Sigma^2\} = s_n M^2$ in $C_{M\kappa}$ and C_{b1}, with M bounded, leading to $\|\mathcal{C}(\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*)\|_F = \mathcal{O}_P(s_n^{2.5}\sqrt{\ln(p_n)/n})$. □

Now we vectorize (4), using $\boldsymbol{\theta} = \text{bvec}(\mathbf{\Delta}) \in \mathbb{R}^{m^2 p_n^2}$, as (cf. (54) in Appendix A)

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^\top(\hat{\mathbf{\Sigma}}_y \boxtimes \hat{\mathbf{\Sigma}}_x)\boldsymbol{\theta} - \boldsymbol{\theta}^\top \text{bvec}(\hat{\mathbf{\Sigma}}_x - \hat{\mathbf{\Sigma}}_y) \tag{39}$$

where previous $L(\mathbf{\Delta}, \hat{\mathbf{\Sigma}}_x, \hat{\mathbf{\Sigma}}_y)$ $(= L(\mathbf{\Delta}))$ is now $\mathcal{L}(\boldsymbol{\theta})$. To include sparse-group penalty, recall that the submatrix $\mathbf{\Delta}^{(k\ell)}$ of $\mathbf{\Delta}$ corresponds to the edge $\{k, \ell\}$ of the MA graph. We denote its vectorized version as $\boldsymbol{\theta}_{Gt} \in \mathbb{R}^{m^2}$ (subscript G for grouped variables, as in [15]) with index $t = 1, 2, \cdots, p_n^2$. Then $\boldsymbol{\theta}_{Gt} = \text{vec}(\mathbf{\Delta}^{(k\ell)})$ where $t = (k-1)p_n + \ell$, $\ell = t \mod p_n$, and $k = \lfloor t/p_n \rfloor + 1$. Using this notation, the penalty $\lambda \sum_{k,\ell=1}^{p_n} \|\mathbf{\Delta}^{(k\ell)}\|_F = \lambda \sum_{t=1}^{p_n^2} \|\boldsymbol{\theta}_{Gt}\|_2$. We now state some restricted strong convexity (RSC) [35], [40], [41] results regarding $\mathcal{L}(\boldsymbol{\theta})$ in Lemma 1. Lemma 1(i) deals with the behavior of $\boldsymbol{\theta}$ centered on $\boldsymbol{\theta}^*$ and it implies the RSC in the sense of [40]. Lemma 1(ii) deals with the behavior of $\boldsymbol{\theta}$ in the subspace consisting of all $\boldsymbol{\theta}$'s such that the support of $\boldsymbol{\theta}$, $\text{supp}(\boldsymbol{\theta}) \subseteq \text{supp}(\boldsymbol{\theta}^*)$, and it implies the RSC in the sense of [35].

The proof of Lemma 1 is given in Appendix B.

Lemma 1: (i) Let $\boldsymbol{\theta}^* = \text{bvec}(\mathbf{\Delta}^*)$, $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\theta}^*$ and $\phi^*_{min} = \phi_{min}(\mathbf{\Sigma}_x^*)\phi_{min}(\mathbf{\Sigma}_y^*)$. Then with probability $> 1 - 2/p_n^{\tau-2}$, for any $\tau > 2$ and $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{m^2 p_n^2}$, we have

$$\tilde{\boldsymbol{\theta}}^\top \hat{\mathbf{\Gamma}} \tilde{\boldsymbol{\theta}} \geq \frac{3}{4}\phi^*_{\min}\|\tilde{\boldsymbol{\theta}}\|_2^2 \tag{40}$$

if $n > N_2$ where

$$N_2 = \max\left\{\frac{1}{M^2}, \left(\frac{192 M s_n}{\phi^*_{min}}\right)^2\right\} C_0^2 \ln(p_n)\,. \tag{41}$$

(ii) Let $\boldsymbol{\theta}_S = \text{bvec}(\mathbf{\Delta}_S) \in \mathbb{R}^{m^2 s_n}$ where $\mathbf{\Delta}_S$ denotes the submatrix of $\mathbf{\Delta}$ with block rows and columns indexed by S, i.e., $\mathbf{\Delta}_S = [\mathbf{\Delta}^{(k\ell)}]_{(k,\ell)\in S}$, and $s_n = |S|$. Then with probability $> 1 - 2/p_n^{\tau-2}$, for any $\tau > 2$ we have

$$\boldsymbol{\theta}_S^\top \hat{\mathbf{\Gamma}}_{S,S} \boldsymbol{\theta}_S \geq \frac{63}{64}\phi^*_{\min}\|\boldsymbol{\theta}_S\|_2^2 \tag{42}$$

for any $\boldsymbol{\theta}_S \in \mathbb{R}^{m^2 s_n}$, if $n > N_2$. •

In Lemma 1(i), the support of $\boldsymbol{\theta}^* = \text{bvec}(\mathbf{\Delta}^*)$, $\text{supp}(\mathbf{\Delta}^*)$, is confined to S whereas that of $\tilde{\boldsymbol{\theta}}$ is not. That is, $\boldsymbol{\theta}^*$ has no more that $m^2 s_n$ nonzero elements whereas none of the elements of $\tilde{\boldsymbol{\theta}}$ need be zero. In Lemma 1(ii), $\boldsymbol{\theta}_S$ has only $m^2 s_n$ elements. One may rewrite

$$\boldsymbol{\theta}_S^\top \hat{\mathbf{\Gamma}}_{S,S} \boldsymbol{\theta}_S = \boldsymbol{\theta}^\top \hat{\mathbf{\Gamma}} \boldsymbol{\theta}, \quad \text{supp}(\boldsymbol{\theta}) \subseteq \text{supp}(\boldsymbol{\theta}^*)\,.$$

That is, while $\hat{\mathbf{\Gamma}}$ is only positive semi-definite, by (42) $\hat{\mathbf{\Gamma}}_{S,S}$ is positive definite. Lemma 1(i) helps in proving that $\mathcal{L}(\boldsymbol{\theta})$ (and hence $\mathcal{L}_\lambda(\boldsymbol{\theta})$) is bounded from below and coercive, hence $\mathcal{L}_\lambda(\boldsymbol{\theta})$ $(= L_\lambda(\mathbf{\Delta}))$ has a minimum in the interior [42, Sec. 2.1], so that analyzing (36) in Theorem 1 makes sense. Lemma 1(ii) is instrumental to proving Theorem 2, stated in the sequel, where it is shown that $\hat{\mathbf{\Delta}}$ of Theorem 1 is a unique minimum of $L_\lambda(\mathbf{\Delta})$.

Setting $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} + \boldsymbol{\theta}^*$, (39) may be rewritten as

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\tilde{\boldsymbol{\theta}}^\top \hat{\mathbf{\Gamma}} \tilde{\boldsymbol{\theta}} + \tilde{\boldsymbol{\theta}}^\top \tilde{\boldsymbol{b}} + \boldsymbol{c} \tag{43}$$

where

$$\tilde{\boldsymbol{b}} = \hat{\mathbf{\Gamma}}\boldsymbol{\theta}^* + \boldsymbol{b}, \tag{44}$$

$$\boldsymbol{b} = \text{bvec}(\hat{\mathbf{\Sigma}}_x - \hat{\mathbf{\Sigma}}_y), \tag{45}$$

$$\boldsymbol{c} = \frac{1}{2}\boldsymbol{\theta}^{*\top}\hat{\mathbf{\Gamma}}\boldsymbol{\theta}^* - \boldsymbol{\theta}^{*\top}\boldsymbol{b}\,. \tag{46}$$

It then follows that

$$\mathcal{L}(\boldsymbol{\theta}) \geq \frac{3}{8}\phi^*_{\min}\|\tilde{\boldsymbol{\theta}}\|_2^2 - \|\tilde{\boldsymbol{b}}\|_2\|\tilde{\boldsymbol{\theta}}\|_2 - \|\boldsymbol{c}\|_2 \tag{47}$$

$$\geq -\frac{2}{3\phi^*_{\min}}\|\tilde{\boldsymbol{b}}\|_2 - \|\boldsymbol{c}\|_2, \tag{48}$$

implying that $\mathcal{L}(\boldsymbol{\theta})$ $(= L(\mathbf{\Delta}))$ is bounded from below. Since the penalty $\rho_\lambda(u) \geq 0$, $\mathcal{L}_\lambda(\boldsymbol{\theta})$ $(= L_\lambda(\mathbf{\Delta}))$ is also bounded from below. By (47), $\lim_{\|\tilde{\boldsymbol{\theta}}\|_2 \to \infty} \mathcal{L}(\boldsymbol{\theta}) \to \infty$, and since $\|\tilde{\boldsymbol{\theta}}\|_2 \leq \|\boldsymbol{\theta}\|_2 + \|\boldsymbol{\theta}^*\|_2$, we have $\lim_{\|\boldsymbol{\theta}\|_2 \to \infty} \mathcal{L}(\boldsymbol{\theta}) \to \infty$, hence coercive. Since $\rho_\lambda(u)$ is non-decreasing on $\mathbb{R}_+$, $\mathcal{L}_\lambda(\boldsymbol{\theta}))$ is bounded from below and coercive, hence $\mathcal{L}_\lambda(\boldsymbol{\theta})$ $(=L_\lambda(\mathbf{\Delta}))$ has a minimum in the interior [42, Sec. 2.1].

Theorem 2 is proved in Appendix B.

Theorem 2: Under Theorem 1, if $n > N_2$ and

$$\phi_{\min}(\mathbf{\Sigma}_y^*)\phi_{\min}(\mathbf{\Sigma}_x^*) > \begin{cases} 0 & : \text{ lasso} \\ \frac{64}{63} \times \frac{1}{a-1} & : \text{ SCAD} \\ \frac{64}{63} \times \frac{\lambda_n}{\epsilon} & : \text{ log-sum,} \end{cases} \tag{49}$$

then with probability $> 1 - 2/p_n^{\tau-2}$, $\tau > 2$, $\hat{\mathbf{\Delta}}$ of Theorem 1 is a unique minimizer of $L_\lambda(\mathbf{\Delta})$. •

Remark 2: The proof of Theorem 2 relies on the fact that under (49), $L(\mathbf{\Delta}_S)$ $(=\mathcal{L}(\boldsymbol{\theta}_S))$ is strictly convex in $\mathbf{\Delta}_S$ (equivalently, in $\boldsymbol{\theta}_S$). We see that as $n \to \infty$, $\lambda_n \to 0$ (see also Remark 1), therefore, we eventually have convexity for log-sum penalty by (49) regardless of the value of $\phi_{\min}(\mathbf{\Sigma}_y^*)\phi_{\min}(\mathbf{\Sigma}_x^*)$ (assuming the latter does not change with p_n). But such is not necessarily the case for SCAD. For SCAD one may need a to become large in which case it would behave more like lasso. □

Remark 3: Here we compare our theoretical results Theorems 1 and 2 with Theorem 1 of [15]. Theorem 1 of this paper is analogous to [15, Theorem 1] where the latter was established for the global optimum of the lasso penalized objective function, whereas Theorem 1 of this paper holds for any stationary point of our non-convex penalized objective function $L_\lambda(\mathbf{\Delta})$. The novelty lies in the proof modifications to handle non-convex penalties. Theorem 2 establishes conditions under which the stationary point analyzed in Theorem 1 of this paper is indeed a unique minimizer of $L_\lambda(\mathbf{\Delta})$ (this result is not needed in [15] as the penalized objective function is convex). □

TABLE 1. ER Graph: F_1 scores, Hamming distances, normalized Frobenius norm of estimation error ($\|\hat{\Delta} - \Delta^*\|_F / \|\Delta^*\|_F$) and timing, for the synthetic data example ($p = 100$, $m = 4$), averaged over 100 runs (standard deviation σ in parentheses). The BIC method is from [15, Sec. III-E].

n	200	400	800	1600
F_1 score (σ): λ's picked to maximize F_1				
Lasso	0.549 (0.099)	0.732 (0.078)	0.865 (0.063)	0.963 (0.036)
Log-sum	0.591 (0.072)	0.759 (0.073)	0.908 (0.050)	0.981 (0.017)
SCAD	0.436 (0.063)	0.642 (0.066)	0.831 (0.066)	0.956 (0.040)
Hamming distance (σ): λ's picked to maximize F_1				
Lasso	189.2 (32.9)	130.7 (33.0)	64.1 (29.9)	18.5 (17.6)
Log-sum	203.8 (41.5)	120.7 (32.6)	44.3 (23.8)	09.5 (08.5)
SCAD	290.2 (44.2)	193.7 (36.2)	81.5 (32.1)	21.6 (19.2)
Est. error (σ): λ's picked to maximize F_1				
Lasso	0.954 (0.022)	0.895 (0.035)	0.856 (0.043)	0.764 (0.054)
Log-sum	0.927 (0.037)	0.844 (0.045)	0.762 (0.057)	0.587 (0.073)
SCAD	1.055 (0.049)	0.879 (0.049)	0.714 (0.070)	0.552 (0.050)
Timing (s) (σ): λ's picked to maximize F_1				
Lasso	06.44 (0.033)	06.62 (0.583)	06.50 (0.258)	06.60 (0.355)
Log-sum	08.26 (0.690)	07.42 (0.673)	07.19 (1.139)	10.94 (1.607)
SCAD	16.45 (0.096)	16.84 (1.007)	16.55 (0.538)	16.81 (0.695)
F_1 score σ): λ's picked to minimize BIC				
Log-sum	0.349 (0.099)	0.699 (0.074)	0.880 (0.100)	0.980 (0.065)
Hamming distance σ): λ's picked to minimize BIC				
Log-sum	990.5 (500.4)	200.4 (97.3)	52.7 (37.8)	8.3 (23.5)

TABLE 2. BA Graph: F_1 scores, Hamming distances and normalized Frobenius norm of estimation error ($\|\hat{\Delta} - \Delta^*\|_F / \|\Delta^*\|_F$) and timing, for the synthetic data example ($p = 100$, $m = 4$), averaged over 100 runs (standard deviation σ in parentheses). The BIC method is from [15, Sec. III-E].

n	200	400	800	1600
F_1 score (σ): λ's picked to maximize F_1				
Lasso	0.665 (0.082)	0.776 (0.050)	0.859 (0.071)	0.949 (0.034)
Log-sum	0.693 (0.078)	0.808 (0.049)	0.896 (0.046)	0.989 (0.011)
SCAD	0.555 (0.069)	0.682 (0.051)	0.806 (0.071)	0.932 (0.038)
Hamming distance (σ): λ's picked to maximize F_1				
Lasso	163.0 (34.7)	125.7 (33.6)	67.2 (31.5)	24.7 (15.9)
Log-sum	150.4 (33.4)	101.5 (27.1)	57.0 (28.8)	05.7 (05.3)
SCAD	236.6 (34.5)	196.4 (46.5)	96.1 (32.2)	33.4 (18.2)
Est. error (σ): λ's picked to maximize F_1				
Lasso	0.955 (0.033)	0.834 (0.041)	0.840 (0.050)	0.774 (0.055)
Log-sum	0.937 (0.047)	0.787 (0.049)	0.647 (0.057)	0.466 (0.046)
SCAD	0.953 (0.049)	0.775 (0.046)	0.716 (0.074)	0.593 (0.069)
Timing (s) (σ): λ's picked to maximize F_1				
Lasso	07.32 (0.185)	07.29 (0.138)	07.22 (0.205)	06.27 (0.434)
Log-sum	08.45 (0.323)	07.90 (0.204)	09.65 (2.716)	10.98 (0.381)
SCAD	18.33 (0.305)	18.28 (0.250)	18.22 (0.304)	17.27 (0.483)
F_1 score σ): λ's picked to minimize BIC				
Log-sum	0.455 (0.078)	0.732 (0.065)	0.852 (0.090)	0.917 (0.139)
Hamming distance σ): λ's picked to minimize BIC				
Log-sum	524.1 (238.9)	148.3 (48.7)	64.0 (34.5)	31.7 (48.8)

V. NUMERICAL EXAMPLES

We now present numerical results for synthetic as well as real data to illustrate the proposed non-convex penalty approaches. In the synthetic data examples the ground truth is known and this allows for an assessment of the efficacy of various approaches. In the real data example the ground truth is unknown and our goal there is visualization and exploration of the differential conditional dependency structures underlying the data. As noted in Sec. III-C, for all numerical results we picked $i_{\max} = 200$ and $\delta = 10^{-3}$ in Algorithms 1 and 2. For the SCAD penalty $a = 3.7$ (as in [4], [25]) and for log-sum penalty $\epsilon = 0.001$. Only λ was treated as a tuning parameter.

A. SYNTHETIC DATA: ERDÖS-RÈNYI AND BARABÁSI-ALBERT GRAPHS

We consider two types of graphs: Erdös-Rènyi (ER) graph and Barabási-Albert (BA) graph [43], [44]. In the ER graph, $p = 100$ nodes are connected to each other with probability $p_{er} = 0.5$ and there are $m = 4$ attributes per node whereas in the BA graph, we used $p = 100$ and mean degree of 2 to generate a BA graph using the procedure given in [44].

In the upper triangular $\mathbf{\Omega}_x$, we set $[\mathbf{\Omega}_x^{(jk)}]_{st} = 0.5^{|s-t|}$ for $j = k = 1, \cdots, p$, $s, t = 1, \cdots, m$. For $j \neq k$, if the two nodes are not connected in the graph (ER or BA), we have $\mathbf{\Omega}^{(jk)} = \mathbf{0}$, and if nodes j and k are connected, then $[\mathbf{\Omega}^{(jk)}]_{st}$ is uniformly distributed over $[-0.4, -0.1] \cup [0.1, 0.4]$ for $s \neq t$, otherwise it is zero. Then add lower triangular elements to make $\mathbf{\Omega}_x$ a symmetric matrix. To generate $\mathbf{\Omega}_y$, we follow [7] and first generate a differential graph with $\mathbf{\Delta} \in \mathbb{R}^{(mp)\times(mp)}$ as an ER graph (regardless of whether $\mathbf{\Omega}_x$ is based on ER or BA model), with connection probability $p_{er} = 0.05$ (sparse): if nodes j and k are connected in the $\mathbf{\Omega}_x$ model, then each of m^2 elements of $\mathbf{\Delta}^{(jk)}$ is independently set to ± 0.9 with equal probabilities. Then $\mathbf{\Omega}_y = \mathbf{\Omega}_x + \mathbf{\Delta}$. Finally add $\gamma \boldsymbol{I}$ to $\mathbf{\Omega}_y$ and to $\mathbf{\Omega}_x$ and pick γ so that $\mathbf{\Omega}_y$ and $\mathbf{\Omega}_x$ are both positive definite. With $\mathbf{\Phi}_x \mathbf{\Phi}_x^\top = \mathbf{\Omega}_x^{-1}$, we generate $\boldsymbol{x} = \mathbf{\Phi} \boldsymbol{w}$ with $\boldsymbol{w} \in \mathbb{R}^{mp}$ as zero-mean Gaussian, with identity covariance, and similarly for $\boldsymbol{y}$. We generate $n = n_x = n_y$ i.i.d. observations for $\boldsymbol{x}$ and $\boldsymbol{y}$, with $m = 4$, $p = 100$, $n \in \{200, 400, 800, 1600\}$.

Simulation results based on 100 runs are shown in Tables 1 and 2 for ER and BA graphs, respectively, where the performance measure are F_1-score and Hamming distance (between estimated and true edgesets $\hat{\mathcal{E}}_{\mathbf{\Delta}}$ and $\mathcal{E}_{\mathbf{\Delta}^*}$) for efficacy in edge detection, normalized estimation error $\|\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*\|_F / \|\mathbf{\Delta}^*\|_F$ and execution time (based on tic-toc functions in MATLAB). All algorithms were run on a Window 10 Pro operating system with processor Intel(R) Core(TM) i7-10700 CPU @2.90 GHz with 32 GB RAM, using MATLAB R2023a. For lasso and SCAD (a=3.7) penalties we used Algorithm 2 whereas for log-sum penalty ($\epsilon = 0.001$) we used LLA-based PGD approach of Algorithm 1 since Algorithm 2 did not work for log-sum penalty as the Lipschitz constant L_c is too high resulting in extremely small step-sizes offering little improvement over lasso. For SCAD, Algorithm 2 yielded better results compared to LLA-based PGD method of Algorithm 1. It is seen that log-sum penalty outperforms lasso and SCAD with F_1 score as the performance metric. For $n \geq 800$, SCAD yields smaller estimation errors in estimating $\mathbf{\Delta}$ for ER graphs in Table 1 but its performance in terms of F_1 score, Hamming distance and execution time metrics is, in general, poor. In practice we do not know the ground truth, hence cannot pick λ to maximize the F_1 score. In Tables 1 and 2 we also show results for log-sum penalty when λ is picked based on a BIC method given in [15, Sec. III-E] and discussed in Sec. III-C.

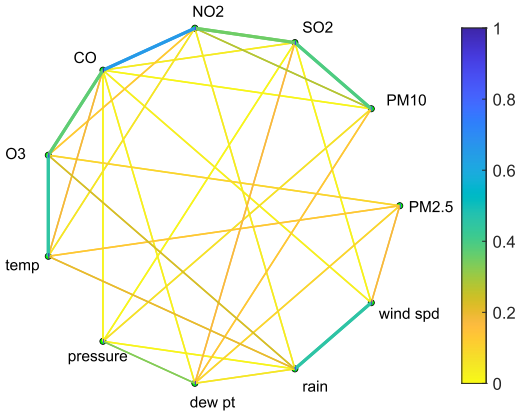
B. REAL DATA: BEIJING AIR-QUALITY DATASET

Here we consider Beijing air-quality dataset [45], [46], downloaded from https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data. This data set includes hourly air pollutants data from 12 nationally-controlled air-quality monitoring sites in the Beijing area from the Beijing Municipal Environmental Monitoring Center, and meteorological data in each air-quality site are matched with the nearest weather station from the China Meteorological Administration. The time period is from March 1st, 2013 to February 28th, 2017. The six air pollutants are $PM_{2.5}$, PM_{10}, SO_2, NO_2, CO, and O_3, and the meteorological data is comprised of five features: temperature, atmospheric pressure, dew point, wind speed, and rain; we did not use wind direction. Thus we have eleven features. We used data from 8 sites: 4 suburban/rural sites – Changping, Huairou, Shunyi, Dingling, and 4 urban area stations – Aotizhongxin, Dongsi, Guanyuan, Gucheng [46, Fig. 1]. The data are averaged over 24 hour period to yield daily averages. We used one year of daily data resulting in $n_x = n_y = 365$ days. The stations are used as attributes, with $m = 4$, for comparison between suburban/rural sites and urban sites using 2013-14 year data.

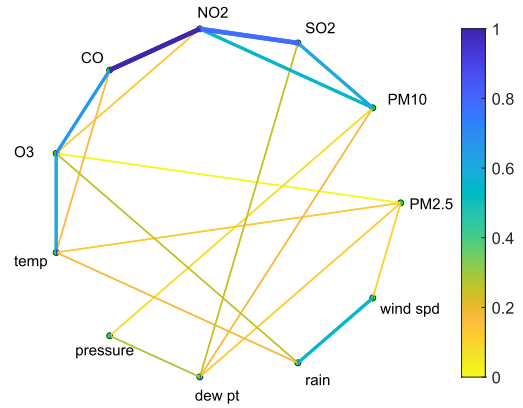
We pre-process the data as follows. Given ith feature data $z_i(t) \in \mathbb{R}^m$, we transform it to $\bar{z}_i(t) = \ln(z_i(t)/z_i(t-1))$ and then detrend it (i.e., remove the best straight-line fit using the MATLAB function detrend). Finally, we scale the detrended scalar sequence to have a mean-square value of one over n_x or n_y samples. The logarithmic transformation and detrending of each feature sequence makes the sequence closer to (univariate) stationary and Gaussian, while scaling "balances" the possible wide variations in the scale of various feature measurements. All temperatures were converted from Celsius to Kelvin to avoid negative numbers, and if a value of a feature is zero (e.g., wind speed), we added a small positive number to it, so that the logarithmic transformation is well-defined.

Fig. 1 shows the estimated differential graphs when comparing daily-averaged data over the period 2013-14, from four suburban/rural sites (x-data) to that from four urban sites (y-data), with air-quality and meteorological variables as $p = 11$ features measured at two sets of 4 monitoring sites (m=4). We used the PGD approach of Algorithm 1 for log-sum penalty. Model selection was done as in [15, Sec. III-E] and in Sec. III-C. The objective is to visualize and explore differential conditional dependency relationships among the 11 variables, comparing one subregion to another. There are significant differences in meteorological conditions and pollutant sources, levels and mutual interactions, among suburban and urban areas [45], [46]. The suburban areas (located toward north) are less polluted than the urban areas (located toward south) [45], [46]. Automobile exhaust is the main cause of NO_2 which is likely to undergo a chemical reaction with Ozone O_3, thereby, lowering its concentration [46]. Cold, dry air from the north reduces both dew point and $PM_{2.5}$ particle concentration in suburban areas while southerly wind brings warmer and more humid air from the more polluted south that elevates the $PM_{2.5}$ concentration [45]. The urban stations neighbor the south of Beijing which is heavily installed with iron, steel and cement industries in Hebei province [45].

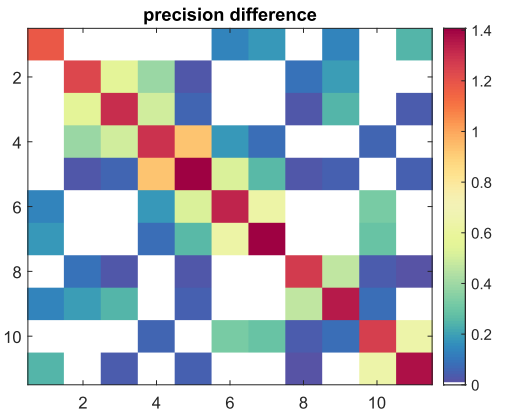
Figs. 1(a)-(d) show estimated $\|\hat{\mathbf{\Delta}}^{(k\ell)}\|_F$ for various edges $\{k, \ell\}$, where it is unscaled in Fig. 1(c),(d) but scaled in Fig. 1(a),(b) so that the largest $\|\hat{\mathbf{\Delta}}^{(k\ell)}\|_F$ (including $k = \ell$)



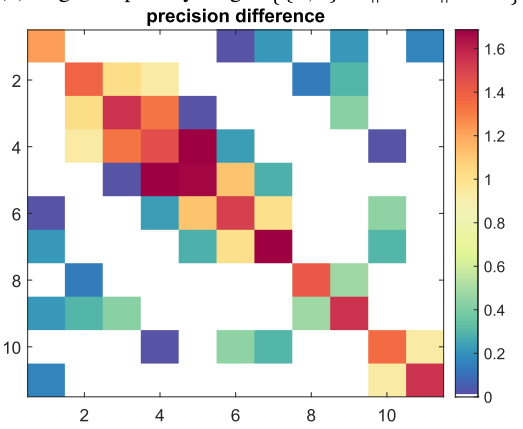
(a) Lasso penalty: edges $\{\{k,\ell\} : \|\hat{\mathbf{\Delta}}^{(k\ell)}\|_F > 0\}$



(b) Log-sum penalty: edges $\{\{k,\ell\} : \|\hat{\mathbf{\Delta}}^{(k\ell)}\|_F > 0\}$



(c) Lasso penalty: $\|\hat{\mathbf{\Delta}}^{(k\ell)}\|_F, k,\ell \in [11]$



(d) Log-sum penalty: $\|\hat{\mathbf{\Delta}}^{(k\ell)}\|_F, k,\ell \in [11]$

FIGURE 1. Differential graphs comparing Beijing air-quality datasets [45] acquired from two sets of monitoring stations, 4 stations per set, year 2013-14: 4 monitoring stations and 11 features ($m = 4$, $p = 11$, $n_X = n_Y = 365$). Number of distinct edges = 35 and 21 in graphs (a) and (b), respectively. Estimated $\|\hat{\Delta}^{(ij)}\|_F$ is the edge weight (normalized to have $\max_{i,j} \|\hat{\Delta}^{(ij)}\|_F = 1$). The edge weights are color coded, in addition to the edges with higher weights being drawn thicker.

is normalized to one. It is seen that the non-convex log-sum penalty yields a much sparser differential graph.

VI. CONCLUSION

A penalized D-trace loss function approach for differential graph learning from multi-attribute data was presented in [15] using convex group-lasso penalty. In this paper we extended [15] to non-convex log-sum and SCAD penalties. Two proximal gradient descent methods were presented to optimize the objective function. Theoretical analysis establishing sufficient conditions for consistency in support recovery, convexity and estimation in high-dimensional settings was provided in Theorems 1 and 2. Numerical results based on synthetic and real data were presented to illustrate the proposed approaches. In the synthetic data examples the log-sum penalized D-trace loss significantly outperformed the lasso-penalized D-trace loss as well as SCAD penalized D-trace loss with F_1-score and Hamming distance as performance metrics.

APPENDIX A TECHNICAL LEMMAS AND PROOF OF THEOREM 1

In this Appendix, we provide a proof of Theorem 1. A first-order necessary condition for minimization of non-convex $L_\lambda(\mathbf{\Delta}, \hat{\mathbf{\Sigma}}_x, \hat{\mathbf{\Sigma}}_y)$ $(=L_\lambda(\mathbf{\Delta}))$, given by (5), w.r.t. $\mathbf{\Delta} \in \mathbb{R}^{mp\times mp}$ is that the zero matrix belongs to the sub-differential of $L_\lambda(\mathbf{\Delta})$ at the solution $\hat{\mathbf{\Delta}}$. That is,

$$\begin{aligned}\mathbf{0} &= \frac{\partial L(\mathbf{\Delta})}{\partial \mathbf{\Delta}} + \lambda \boldsymbol{Z}(\mathbf{\Delta})\Big|_{\mathbf{\Delta}=\hat{\mathbf{\Delta}}} \\ &= \hat{\mathbf{\Sigma}}_x \hat{\mathbf{\Delta}} \hat{\mathbf{\Sigma}}_y - (\hat{\mathbf{\Sigma}}_x - \hat{\mathbf{\Sigma}}_y) + \lambda \boldsymbol{Z}(\hat{\mathbf{\Delta}}) \end{aligned} \tag{50}$$

where $\lambda \boldsymbol{Z}(\mathbf{\Delta}) \in \partial \sum_{k,\ell=1}^{p} \rho_\lambda\big(\|\mathbf{\Delta}^{(k\ell)}\|_F\big) \in \mathbb{R}^{mp\times mp}$, the sub-differential of (possibly non-convex) penalty term, is given by

$$(\boldsymbol{Z}(\mathbf{\Delta}))^{(k\ell)} = \begin{cases} \boldsymbol{V} \in \mathbb{R}^{m\times m},\ \|\boldsymbol{V}\|_F \le 1, \\ \quad \text{if } \|\mathbf{\Delta}^{(k\ell)}\|_F = 0 \\ \frac{\mathbf{\Delta}^{(k\ell)}}{\|\mathbf{\Delta}^{(k\ell)}\|_F} \text{ if } \|\mathbf{\Delta}^{(k\ell)}\|_F \neq 0 \ :\ \text{lasso} \\ C^{(k\ell)} \text{ if } \|\mathbf{\Delta}^{(k\ell)}\|_F \neq 0 \ :\ \text{log-sum} \\ D^{(k\ell)} \text{ if } \|\mathbf{\Delta}^{(k\ell)}\|_F \neq 0 \ :\ \text{SCAD}, \end{cases} \tag{51}$$

$$C^{(k\ell)} = \frac{\epsilon}{\epsilon + \|\boldsymbol{\Delta}^{(k\ell)}\|_F} \frac{\boldsymbol{\Delta}^{(k\ell)}}{\|\boldsymbol{\Delta}^{(k\ell)}\|_F}, \tag{52}$$

$$D^{(k\ell)} = \begin{cases} \frac{\boldsymbol{\Delta}^{(k\ell)}}{\|\boldsymbol{\Delta}^{(k\ell)}\|_F} \text{ if } 0 < \|\boldsymbol{\Delta}^{(k\ell)}\|_F \le \lambda \\ \frac{a - \|\boldsymbol{\Delta}^{(k\ell)}\|_F/\lambda}{a-1} \frac{\boldsymbol{\Delta}^{(k\ell)}}{\|\boldsymbol{\Delta}^{(k\ell)}\|_F} \\ \qquad \text{if } \lambda < \|\boldsymbol{\Delta}^{(k\ell)}\|_F \le a\lambda \\ \mathbf{0} \text{ if } a\lambda < \|\boldsymbol{\Delta}^{(k\ell)}\|_F . \end{cases} \tag{53}$$

We have $\|(\boldsymbol{Z}(\boldsymbol{\Delta}))^{(k\ell)}\|_F = \|\text{vec}((\boldsymbol{Z}(\boldsymbol{\Delta}))^{(k\ell)})\|_2 \le 1$ for all three penalties. In the case of the lasso penalty, this property was used to prove [15, Theorem 1] to establish consistency in support recovery and estimation for the global minimum $\hat{\boldsymbol{\Delta}}$ of $L_\lambda(\boldsymbol{\Delta})$. With non-convex penalties we have only a local minimum $\hat{\boldsymbol{\Delta}}$ satisfying (50) with such properties. In the rest of the section we provide a proof of Theorem 1 for non-convex penalties. This requires recalling some of the developments from [15, Appendix A].

In terms of $m \times m$ submatrices of $\boldsymbol{\Delta}$, $\hat{\boldsymbol{\Sigma}}_x$, $\hat{\boldsymbol{\Sigma}}_y$ and $\boldsymbol{Z}(\boldsymbol{\Delta})$ corresponding to various graph edges, using $\text{bvec}(\boldsymbol{ADB}) = (\boldsymbol{B}^\top \boxtimes \boldsymbol{A})\text{bvec}(\boldsymbol{D})$ [32, Lemma 1], we may rewrite (50) as

$$(\hat{\boldsymbol{\Sigma}}_y \boxtimes \hat{\boldsymbol{\Sigma}}_x)\text{bvec}(\hat{\boldsymbol{\Delta}}) - \text{bvec}(\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y) + \lambda\, \text{bvec}(\boldsymbol{Z}(\hat{\boldsymbol{\Delta}})) = \mathbf{0} \tag{54}$$

Then (54) can be rewritten as

$$\begin{bmatrix} \hat{\boldsymbol{\Gamma}}_{S,S} & \hat{\boldsymbol{\Gamma}}_{S,S^c} \\ \hat{\boldsymbol{\Gamma}}_{S^c,S} & \hat{\boldsymbol{\Gamma}}_{S^c,S^c} \end{bmatrix} \begin{bmatrix} \text{bvec}(\hat{\boldsymbol{\Delta}}_S) \\ \text{bvec}(\hat{\boldsymbol{\Delta}}_{S^c}) \end{bmatrix} - \begin{bmatrix} \text{bvec}((\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y)_S) \\ \text{bvec}((\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y)_{S^c}) \end{bmatrix} + \lambda \begin{bmatrix} \text{bvec}(\boldsymbol{Z}(\hat{\boldsymbol{\Delta}}_S)) \\ \text{bvec}(\boldsymbol{Z}(\hat{\boldsymbol{\Delta}}_{S^c})) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{55}$$

The general approach of [5] (followed in [7], [13], [28], [29]) is to first solve the hypothetical constrained optimization problem with known edgeset S

$$\tilde{\boldsymbol{\Delta}} = \arg \min_{\boldsymbol{\Delta}: \boldsymbol{\Delta}_{S^c} = \mathbf{0}} L_\lambda(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) \tag{56}$$

where S^c is the complement of S. Since, by construction, $\tilde{\boldsymbol{\Delta}}_{S^c} = \mathbf{0}$, in this case (55) reduces to

$$\hat{\boldsymbol{\Gamma}}_{S,S}\text{bvec}(\tilde{\boldsymbol{\Delta}}_S) - \text{bvec}((\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y)_S) + \lambda\, \text{bvec}(\boldsymbol{Z}(\tilde{\boldsymbol{\Delta}}_S)) = \mathbf{0}. \tag{57}$$

In the approach of [5], one investigates conditions under which the solution $\hat{\boldsymbol{\Delta}}$ to (5) is the same as the solution $\tilde{\boldsymbol{\Delta}}$ to (56). This is done by showing that $\hat{\boldsymbol{\Delta}}$ satisfies (55). The choice $\hat{\boldsymbol{\Delta}} = \tilde{\boldsymbol{\Delta}}$ implies that $\hat{\boldsymbol{\Delta}}_{S^c} = \mathbf{0}$ and (57) is true with $\tilde{\boldsymbol{\Delta}}$ replaced with $\hat{\boldsymbol{\Delta}}$. In order to satisfy (55), it remains to show that for any edge $e \in S^c$, we have strict feasibility

$$\|\hat{\boldsymbol{\Gamma}}_{e,S}\text{bvec}(\tilde{\boldsymbol{\Delta}}_S) - \text{bvec}((\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y)_e)\|_2 < \lambda, \tag{58}$$

where for $\boldsymbol{a} \in \mathbb{R}^q$, $\|\boldsymbol{a}\|_2 = \sqrt{\boldsymbol{a}^\top \boldsymbol{a}}$. This requires a set of sufficient conditions stated in Theorem 1.

Lemma 2 [15]: Let $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$ be as in (3), $\bar{\sigma}_{xy}$ as in (34), C_0 as in (35) and assume data are Gaussian. Define $n = \min(n_x, n_y)$ and

$$\mathcal{A} = \max\left\{\|\mathcal{C}(\hat{\boldsymbol{\Sigma}}_x - \boldsymbol{\Sigma}_x^*)\|_\infty, \|\mathcal{C}(\hat{\boldsymbol{\Sigma}}_y - \boldsymbol{\Sigma}_y^*)\|_\infty\right\}. \tag{59}$$

Then for any $\tau > 2$ and $n > 2m^2 \ln(4m^2 p_n^\tau)$,

$$P\left(\mathcal{A} > C_0\sqrt{\ln(p_n)/n}\right) \le 2/p_n^{\tau-2} \quad \bullet \tag{60}$$

Recall (29)-(33) and define

$$\boldsymbol{\Delta}_x = \hat{\boldsymbol{\Sigma}}_x - \boldsymbol{\Sigma}_x^*,\ \boldsymbol{\Delta}_y = \hat{\boldsymbol{\Sigma}}_y - \boldsymbol{\Sigma}_y^*,\ \boldsymbol{\Delta}_\Gamma = \hat{\boldsymbol{\Gamma}} - \boldsymbol{\Gamma}^*, \tag{61}$$

$$\boldsymbol{\Delta}_\Sigma = \boldsymbol{\Delta}_x - \boldsymbol{\Delta}_y,\ \epsilon_{0x} = \|\mathcal{C}(\boldsymbol{\Delta}_x)\|_\infty, \tag{62}$$

$$\epsilon_{0y} = \|\mathcal{C}(\boldsymbol{\Delta}_y)\|_\infty,\ \epsilon_0 > \max\{\epsilon_{0x}, \epsilon_{0y}\}. \tag{63}$$

Lemma 3 [15]: Assume that

$$\kappa_\Gamma < \frac{1}{3s_n(\epsilon_0^2 + 2M\epsilon_0)}. \tag{64}$$

Let ($\boldsymbol{\Gamma}_{S,S}^{-*}$ denotes $(\boldsymbol{\Gamma}_{S,S}^*)^{-1}$)

$$R(\boldsymbol{\Delta}_\Gamma) = \hat{\boldsymbol{\Gamma}}_{S,S}^{-1} - \boldsymbol{\Gamma}_{S,S}^{-*} + \boldsymbol{\Gamma}_{S,S}^{-*}(\boldsymbol{\Delta}_\Gamma)_{S,S}\boldsymbol{\Gamma}_{S,S}^{-*}. \tag{65}$$

Then we have

$$\|\mathcal{C}(R(\boldsymbol{\Delta}_\Gamma))\|_\infty \le \frac{3}{2}\kappa_\Gamma^3 s_n(\epsilon_0^2 + 2M\epsilon_0)^2, \tag{66}$$

$$\|\mathcal{C}(R(\boldsymbol{\Delta}_\Gamma))\|_{1,\infty} \le \frac{3}{2}\kappa_\Gamma^3 s_n^2(\epsilon_0^2 + 2M\epsilon_0)^2, \tag{67}$$

$$\|\mathcal{C}(\hat{\boldsymbol{\Gamma}}_{S,S}^{-1} - \boldsymbol{\Gamma}_{S,S}^{-*})\|_\infty \le \kappa_\Gamma^2(\epsilon_0^2 + 2M\epsilon_0) \times \left(1 + 1.5\, s_n\kappa_\Gamma(\epsilon_0^2 + 2M\epsilon_0)\right), \tag{68}$$

$$\|\mathcal{C}(\hat{\boldsymbol{\Gamma}}_{S,S}^{-1} - \boldsymbol{\Gamma}_{S,S}^{-*})\|_{1,\infty} \le s_n\|\mathcal{C}(\hat{\boldsymbol{\Gamma}}_{S,S}^{-1} - \boldsymbol{\Gamma}_{S,S}^{-*})\|_\infty \quad \bullet \tag{69}$$

Lemma 4 stated and proved below is as in [15] but requires use of $\|(\boldsymbol{Z}(\boldsymbol{\Delta}))^{(k\ell)}\|_F = \|\text{vec}((\boldsymbol{Z}(\boldsymbol{\Delta}))^{(k\ell)})\|_2 \le 1$ where $(\boldsymbol{Z}(\boldsymbol{\Delta}))^{(k\ell)}$ is as in (51).

Lemma 4: Assume (64) and the following conditions:

$$0 < \alpha < 1 \text{ where } \alpha \text{ is as in (33)}, \tag{70}$$

$$\epsilon_0 < \min\left\{M, \frac{\alpha\lambda_n}{2(2-\alpha)}\right\}, \tag{71}$$

$$\alpha C_\alpha \min\{\lambda_n, 1\} \ge 3\, s_n\epsilon_0 M\kappa_\Gamma B_s \tag{72}$$

where

$$C_\alpha = \frac{\alpha\lambda_n + 2\epsilon_0\alpha - 4\epsilon_0}{2M\alpha\lambda_n + \alpha\lambda_n + 2\epsilon_0\alpha}, \tag{73}$$

$$B_s = \left[1 + \kappa_\Gamma\left(3\, s_n\epsilon_0 M + \min\{s_n M^2, M_\Sigma^2\}\right) \times \left(4.5\, s_n\epsilon_0 M\kappa_\Gamma + 1\right)\right]. \tag{74}$$

Then we have

(i) $\text{bvec}(\hat{\boldsymbol{\Delta}}_{S^c}) = \mathbf{0}$.

(ii) $\|\mathcal{C}(\hat{\boldsymbol{\Delta}} - \boldsymbol{\Delta}^*)\|_\infty \le 2\lambda_n\kappa_\Gamma + 3s_n\epsilon_0 M\kappa_\Gamma^2\left(4.5\, s_n\epsilon_0 M\kappa_\Gamma + 1\right)\left(2M + 2\lambda_n\right)$ $\bullet$

Proof: To establish part (i), as in [15, Lemma 4], we need to show that (58) is true. Let d denote the left-side of (58). It follow from (57) that

$$\text{bvec}(\tilde{\boldsymbol{\Delta}}_S) = \hat{\boldsymbol{\Gamma}}_{S,S}^{-1}\left(\text{bvec}((\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y)_S) - \lambda\, \text{bvec}(\boldsymbol{Z}(\tilde{\boldsymbol{\Delta}}_S))\right). \tag{75}$$

Substitute (75) in the left-side of (58) to yield

$$d = \|\hat{\mathbf{\Gamma}}_{e,S}\big[\hat{\mathbf{\Gamma}}_{S,S}^{-1}\big(\text{bvec}((\hat{\mathbf{\Sigma}}_x - \hat{\mathbf{\Sigma}}_y)_S) - \lambda\,\text{bvec}(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S))\big)\big] - \text{bvec}((\hat{\mathbf{\Sigma}}_x - \hat{\mathbf{\Sigma}}_y)_e)\|_2\,. \tag{76}$$

At the true values we have

$$\mathbf{0} = \frac{\partial L(\mathbf{\Delta}, \mathbf{\Sigma}_x^*, \mathbf{\Sigma}_y^*)}{\partial \mathbf{\Delta}}\Big|_{\mathbf{\Delta}=\mathbf{\Delta}^*} = \mathbf{\Sigma}_x^*\mathbf{\Delta}^*\mathbf{\Sigma}_y^* - (\mathbf{\Sigma}_x^* - \mathbf{\Sigma}_y^*)$$

implying

$$\mathbf{\Gamma}^*\text{bvec}(\mathbf{\Delta}^*) - \text{bvec}(\mathbf{\Sigma}_x^* - \mathbf{\Sigma}_y^*) = \mathbf{0}, \tag{77}$$

which, noting that $(\mathbf{\Delta}^*)_{S^c} = \mathbf{0}$, can be rewritten as (cf. (55))

$$\mathbf{\Gamma}_{S,S}^*\text{bvec}(\mathbf{\Delta}_S^*) = \text{bvec}(\mathbf{\Sigma}_x^*)_S - \text{bvec}(\mathbf{\Sigma}_y^*)_S, \tag{78}$$
$$\mathbf{\Gamma}_{e,S}^*\text{bvec}(\mathbf{\Delta}_S^*) = \text{bvec}(\mathbf{\Sigma}_x^*)_e - \text{bvec}(\mathbf{\Sigma}_y^*)_e\,. \tag{79}$$

Therefore, $(\boldsymbol{A}^{-*} = (\boldsymbol{A}^*)^{-1})$,

$$\mathbf{\Gamma}_{e,S}^*\mathbf{\Gamma}_{S,S}^{-*}\big(\text{bvec}(\mathbf{\Sigma}_x^*)_S - \text{bvec}(\mathbf{\Sigma}_y^*)_S\big) - \text{bvec}(\mathbf{\Sigma}_x^*)_e + \text{bvec}(\mathbf{\Sigma}_y^*)_e = \mathbf{0}\,. \tag{80}$$

Recalling (61) and using (80) in (76),

$$d = \|\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1}\text{bvec}((\mathbf{\Delta}_\Sigma)_S) + \big(\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{e,S}^*\mathbf{\Gamma}_{S,S}^{-*}\big)\big(\text{bvec}(\mathbf{\Sigma}_x^*)_S - \text{bvec}(\mathbf{\Sigma}_y^*)_S\big) - \lambda\,\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1}\text{bvec}(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S)) - \text{bvec}((\mathbf{\Delta}_\Sigma)_e)\|_2\,. \tag{81}$$

To bound various terms in (81), using [15, Eq. (80)], we have

$$\|\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1}\text{bvec}((\mathbf{\Delta}_\Sigma)_S)\|_2 \le \|\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1})\|_1\,\|\boldsymbol{\mathcal{C}}(\mathbf{\Delta}_\Sigma)\|_\infty, \tag{82}$$

$$\|\big(\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{e,S}^*\mathbf{\Gamma}_{S,S}^{-*}\big)\big(\text{bvec}(\mathbf{\Sigma}_x^*)_S - \text{bvec}(\mathbf{\Sigma}_y^*)_S\big)\|_2 \le \|\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{e,S}^*\mathbf{\Gamma}_{S,S}^{-*})\|_1\,\|\boldsymbol{\mathcal{C}}(\mathbf{\Sigma}_x^* - \mathbf{\Sigma}_y^*)\|_\infty, \tag{83}$$

$$\|\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1}\text{bvec}(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S))\|_2 \le \|\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1})\|_1\,\|\boldsymbol{\mathcal{C}}(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S))\|_\infty \le \|\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Gamma}}_{e,S}\hat{\mathbf{\Gamma}}_{S,S}^{-1})\|_1, \tag{84}$$

$$\|\text{bvec}((\mathbf{\Delta}_\Sigma)_e)\|_2 \le \|\boldsymbol{\mathcal{C}}(\mathbf{\Delta}_\Sigma)\|_\infty, \tag{85}$$

where in (84) we have used the fact that

$$\|\boldsymbol{\mathcal{C}}(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S))\|_\infty = \max_{\{k,\ell\}\in S}\|(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S))^{(k,\ell)}\|_F \le 1\,.$$

Now mimic the proof of [15, Lemma 4] from [15, Eq. (85)] through [15, Eq. (101)] to conclude that $d < \lambda$, and hence, (58) is true, proving part (i) of Lemma 4.

We now turn to the proof of Lemma 4(ii). Since $\hat{\mathbf{\Delta}} = \tilde{\mathbf{\Delta}}$, for any edge $\{k,\ell\} \in S$, we have

$$\|(\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*)^{(k\ell)}\|_F = \|(\tilde{\mathbf{\Delta}} - \mathbf{\Delta}^*)^{(k\ell)}\|_F = \|\text{vec}(\tilde{\mathbf{\Delta}}^{(k\ell)}) - \text{vec}((\mathbf{\Delta}^*)^{(k\ell)})\|_2\,. \tag{86}$$

Using (57) and (78)

$$\begin{aligned}\text{bvec}((\tilde{\mathbf{\Delta}} - \mathbf{\Delta}^*)_S) &= \hat{\mathbf{\Gamma}}_{S,S}^{-1}\text{bvec}((\mathbf{\Delta}_\Sigma)_S) + (\hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{S,S}^{-*}) \times \text{bvec}((\mathbf{\Sigma}_x^* - \mathbf{\Sigma}_y^*)_S) - \lambda_n\hat{\mathbf{\Gamma}}_{S,S}^{-1}\,\text{bvec}(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S))\,.\end{aligned} \tag{87}$$

Since $\hat{\mathbf{\Gamma}}_{S,S}^{-1} = \hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{S,S}^{-*} + \mathbf{\Gamma}_{S,S}^{-*}$,

$$\|\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Gamma}}_{S,S}^{-1})\|_{1,\infty} \le \|\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{S,S}^{-*})\|_{1,\infty} + \|\boldsymbol{\mathcal{C}}(\mathbf{\Gamma}_{S,S}^{-*})\|_{1,\infty}. \tag{88}$$

By (87), for any edge $f = \{k,\ell\} \in S$, we have

$$\begin{aligned}&\|\text{vec}((\tilde{\mathbf{\Delta}} - \mathbf{\Delta}^*)^{(k\ell)})\|_2 \\ &\le \|(\hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{S,S}^{-*})_{f,S} \times \text{bvec}\big((\mathbf{\Delta}_\Sigma)_S + (\mathbf{\Sigma}_x^* - \mathbf{\Sigma}_y^*)_S - \lambda_n\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S)\big)\|_2 \\ &\quad + \|(\mathbf{\Gamma}_{S,S}^{-*})_{f,S}\,\text{bvec}\Big((\mathbf{\Delta}_\Sigma)_S - \lambda_n\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S)\Big)\|_2 \\ &\le \|(\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{S,S}^{-*}))_{f,S}\|_1\,\Big(\|\boldsymbol{\mathcal{C}}(\mathbf{\Delta}_\Sigma)\|_\infty + \|\boldsymbol{\mathcal{C}}(\mathbf{\Sigma}_x^* - \mathbf{\Sigma}_y^*)\|_\infty + \lambda_n\|\boldsymbol{\mathcal{C}}(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S))\|_\infty\Big) + \|(\boldsymbol{\mathcal{C}}(\mathbf{\Gamma}_{S,S}^{-*}))_{f,S}\|_1\,\big(\|\boldsymbol{\mathcal{C}}(\mathbf{\Delta}_\Sigma)\|_\infty \\ &\quad + \lambda_n\|\boldsymbol{\mathcal{C}}(\boldsymbol{Z}(\tilde{\mathbf{\Delta}}_S))\|_\infty\big) \\ &\le \|\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Gamma}}_{S,S}^{-1} - \mathbf{\Gamma}_{S,S}^{-*})\|_{1,\infty}\,\Big(\|\boldsymbol{\mathcal{C}}(\mathbf{\Delta}_\Sigma)\|_\infty + \|\boldsymbol{\mathcal{C}}(\mathbf{\Sigma}_x^* - \mathbf{\Sigma}_y^*)\|_\infty + \lambda_n\Big) + \|\boldsymbol{\mathcal{C}}(\mathbf{\Gamma}_{S,S}^{-*})\|_{1,\infty}\,\big(\|\boldsymbol{\mathcal{C}}(\mathbf{\Delta}_\Sigma)\|_\infty + \lambda_n\big) \\ &\le s_n\kappa_\Gamma^2(\epsilon_0^2 + 2M\epsilon_0)\big(1 + 1.5\,s_n(\epsilon_0^2 + 2M\epsilon_0)\kappa_\Gamma\big) \times (2\epsilon_0 + 2M + \lambda_n) + \kappa_\Gamma(2\epsilon_0 + \lambda_n) =: U_{b3}\,.\end{aligned} \tag{89}$$

By (71), for $0 < \alpha < 1$, we have $2\epsilon_0 < \alpha\lambda_n/(2-\alpha) < \alpha\lambda_n < \lambda_n$. Therefore, $\kappa_\Gamma(2\epsilon_0 + \lambda_n) < 2\kappa_\Gamma\lambda_n$ and $2\epsilon_0 + 2M + \lambda_n < 2M + 2\lambda_n$. Since $\epsilon_0 < M$ by (71), we also have $\epsilon_0^2 + 2M\epsilon_0 < 3M\epsilon_0$. Using these relations and (89), it follows that

$$U_{b3} \le 3s_n\epsilon_0 M\kappa_\Gamma^2\big(1 + 4.5\,s_n\epsilon_0 M\kappa_\Gamma\big)\big(2M + 2\lambda_n\big) + 2\lambda_n\kappa_\Gamma\,.$$

Finally,

$$\|\boldsymbol{\mathcal{C}}(\hat{\mathbf{\Delta}} - \mathbf{\Delta}^*)\|_\infty = \max_{f=\{k,\ell\}\in S}\|\text{vec}((\tilde{\mathbf{\Delta}} - \mathbf{\Delta}^*)^{(k\ell)})\|_2,$$

proving the desired result. ■

Proof of Theorem 1: With Lemmas 2-4 in place, we simply mimic the proof of [15, Theorem 1]; no changes are needed. ■

APPENDIX B PROOFS OF LEMMA 1 AND THEOREM 2

Proof of Lemma 1: We first prove part (i). Following Eqns. (135)-(137) of [15] we have

$$|\tilde{\boldsymbol{\theta}}^\top(\hat{\mathbf{\Gamma}} - \mathbf{\Gamma}^*)\tilde{\boldsymbol{\theta}}| \le 16\,s_n(\epsilon_0^2 + 2M\epsilon_0)\|\tilde{\boldsymbol{\theta}}\|_2^2\,. \tag{90}$$

We have

$$\tilde{\boldsymbol{\theta}}^\top\hat{\mathbf{\Gamma}}\tilde{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}^\top\mathbf{\Gamma}^*\tilde{\boldsymbol{\theta}} + \tilde{\boldsymbol{\theta}}^\top(\hat{\mathbf{\Gamma}} - \mathbf{\Gamma}^*)\tilde{\boldsymbol{\theta}} \ge \phi_{min}^*\,\|\tilde{\boldsymbol{\theta}}\|_2^2 - 48\,s_nM\epsilon_0\,\|\tilde{\boldsymbol{\theta}}\|_2^2 \tag{91}$$

where we used the fact that $\epsilon_0 < M$, by Lemma 4. Pick ϵ_0 to satisfy

$$\epsilon_0 = C_0\sqrt{ln(p_n)/n} \le \min\big\{M, \frac{\phi_{min}^*}{192 s_nM}\big\}, \tag{92}$$

leading to $48 s_n M\epsilon_0 \le \phi^*_{min}/4$ w.h.p. for $n > N_2$. This proves Lemma 1(i). We now turn to part (ii). With $\boldsymbol{\Delta}_\Gamma = \hat{\boldsymbol{\Gamma}} - \boldsymbol{\Gamma}^*$ and $\boldsymbol{\theta}$ such that supp$(\boldsymbol{\theta}) \subseteq$ supp$(\boldsymbol{\theta}^*)$, we have

$$\boldsymbol{\theta}_S^\top \hat{\boldsymbol{\Gamma}}_{S,S}\boldsymbol{\theta}_S = \boldsymbol{\theta}^\top \hat{\boldsymbol{\Gamma}}\boldsymbol{\theta} = \boldsymbol{\theta}^\top \boldsymbol{\Gamma}^*\boldsymbol{\theta} + \boldsymbol{\theta}^\top \boldsymbol{\Delta}_\Gamma \boldsymbol{\theta}\,. \tag{93}$$

As in Eqn. (135) of [15], we have

$$\boldsymbol{\theta}^\top \boldsymbol{\Delta}_\Gamma \boldsymbol{\theta} = \sum_{t_1=1}^{p^2}\sum_{t_2=1}^{p^2} \boldsymbol{\theta}_{Gt_1}^\top (\boldsymbol{\Delta}_\Gamma)_{Gt_1,Gt_2}\boldsymbol{\theta}_{Gt_2}. \tag{94}$$

Therefore,

$$\begin{aligned}
|\boldsymbol{\theta}^\top \boldsymbol{\Delta}_\Gamma \boldsymbol{\theta}| &\le \sum_{t_1=1}^{p^2}\sum_{t_2=1}^{p^2} |\boldsymbol{\theta}_{Gt_1}^\top (\boldsymbol{\Delta}_\Gamma)_{Gt_1,Gt_2}\boldsymbol{\theta}_{Gt_2}| \\
&\le \sum_{t_1=1}^{p^2}\sum_{t_2=1}^{p^2} \|\boldsymbol{\theta}_{Gt_1}\|_2 \, \|(\boldsymbol{\Delta}_\Gamma)_{Gt_1,Gt_2}\|_F \, \|\boldsymbol{\theta}_{Gt_2}\|_2 \\
&\le \|\boldsymbol{\mathcal{C}}(\boldsymbol{\Delta}_\Gamma)\|_\infty \Big(\sum_{t=1}^{p^2} \|\boldsymbol{\theta}_{Gt}\|_2\Big)^2 \\
&\le (\epsilon_0^2 + 2M\epsilon_0)\, s_n \|\boldsymbol{\theta}\|_2^2 \le 3M\epsilon_0 s_n \|\boldsymbol{\theta}\|_2^2, \qquad (95)
\end{aligned}$$

where we used $\|\boldsymbol{\mathcal{C}}(\boldsymbol{\Delta}_\Gamma)\|_\infty < \epsilon_0^2 + 2M\epsilon_0$ by [15, Eqn. (57)], $\epsilon_0 < M$ (Lemma 4) and the fact that by the Cauchy-Schwarz inequality,

$$\sum_{t=1}^{p^2} \|\tilde{\boldsymbol{\theta}}_{Gt}\|_2 \le \sqrt{s_n}\, \|\boldsymbol{\theta}\|_2\,.$$

Thus, for $\boldsymbol{\theta}$ such that supp$(\boldsymbol{\theta}) \subseteq$ supp$(\boldsymbol{\theta}^*)$,

$$\boldsymbol{\theta}_S^\top \hat{\boldsymbol{\Gamma}}_{S,S}\boldsymbol{\theta}_S \ge \phi^*_{min}\, \|\boldsymbol{\theta}\|_2^2 - 3\, s_n M\epsilon_0\, \|\boldsymbol{\theta}\|_2^2 \ge \frac{63}{64}\phi^*_{\min}\|\boldsymbol{\theta}_S\|_2^2 \tag{96}$$

for $n > N_2$ with ϵ_0 chosen as in part (i) of Lemma 1. ■

Proof of Theorem 2: With $\boldsymbol{\Delta}$ restricted to supp$(\boldsymbol{\Delta}) \subseteq$ supp$(\boldsymbol{\Delta}^*)$, by Lemma 1(ii), $L(\boldsymbol{\Delta}) - \frac{\mu}{2}\|\boldsymbol{\Delta}\|_F^2 = L(\boldsymbol{\Delta}_S) - \frac{\mu}{2}\|\boldsymbol{\Delta}_S\|_F^2$ is strictly convex for $\mu < (63/64)\phi^*_{\min}$ since its Hessian is $\hat{\boldsymbol{\Gamma}}_{S,S} - \mu \boldsymbol{I}_{m^2 s_n}$. By property (v) of the penalty functions, $g(u) := \rho_\lambda(u) + \frac{\mu}{2}u^2$ is convex, for some $\mu \ge 0$ ($\mu = 0$ for lasso, $= 1/(1-a)$ for SCAD, and $= \lambda_n/\epsilon$ for log-sum), and by property (ii), it is non-decreasing on $\mathbb{R}_+$. Therefore, by the composition rules [47, Sec. 3.2.4], $g(\|\boldsymbol{\Delta}_S^{(k\ell)}\|_F)$ is convex. Hence, under (49), $L_\lambda(\boldsymbol{\Delta}_S)$ is strictly convex in $\boldsymbol{\Delta}_S$ and therefore, the solution to (56) via (50)-(55) and (57) yields a unique minimizer. Finally, by Theorem 2(ii), unrestricted $\hat{\boldsymbol{\Delta}}$ of Theorem 1 is unique. ■

REFERENCES

[1] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. Hoboken, NJ, USA: Wiley, 1990.

[2] S. L. Lauritzen, *Graphical Models*. London, U.K.: Oxford Univ. Press, 1996.

[3] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data*. Berlin, Germany: Springer, 2011.

[4] C. Lam and J. Fan, "Sparsistency and rates of convergence in large covariance matrix estimation," *Ann. Statist.*, vol. 37, no. 6, pp. 4254–4278, Dec. 2009.

[5] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, "High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence," *Electron. J. Statist.*, vol. 5, pp. 935–980, Jan. 2011.

[6] M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge, U.K.: Cambridge Univ. Press, 2019.

[7] H. Yuan, R. Xi, C. Chen, and M. Deng, "Differential network analysis via lasso penalized D-trace loss," *Biometrika*, vol. 104, no. 4, pp. 755–770, Dec. 2017.

[8] Z. Tang, Z. Yu, and C. Wang, "A fast iterative algorithm for high-dimensional differential network," *Comput. Statist.*, vol. 35, no. 1, pp. 95–109, Mar. 2020.

[9] Y. Wu, T. Li, X. Liu, and L. Chen, "Differential network inference via the fused D-trace loss with cross variables," *Electron. J. Statist.*, vol. 14, no. 1, pp. 1269–1301, Jan. 2020.

[10] S. D. Zhao, T. T. Cai, and H. Li, "Direct estimation of differential networks," *Biometrika*, vol. 101, no. 2, pp. 253–268, Jun. 2014.

[11] E. Belilovsky, G. Varoquaux, and M. B. Blaschko, "Hypothesis testing for differences in Gaussian graphical models: Applications to brain connectivity," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 29, Dec. 2016, pp. 1–9.

[12] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 76, no. 2, pp. 373–397, Mar. 2014.

[13] M. Kolar, H. Liu, and E. P. Xing, "Graph estimation from multi-attribute data," *J. Mach. Learn. Res.*, vol. 15, pp. 1713–1750, Jan. 2012.

[14] J. K. Tugnait, "Sparse-group lasso for graph learning from multi-attribute data," *IEEE Trans. Signal Process.*, vol. 69, pp. 1771–1786, 2021.

[15] J. K. Tugnait, "Learning high-dimensional differential graphs from multi-attribute data," *IEEE Trans. Signal Process.*, vol. 72, pp. 415–431, 2024.

[16] G. Marjanovic and V. Solo, "Vector ℓ_0 sparse conditional independence graphs," in *Proc. IEEE ICASSP*, Apr. 2018, pp. 2731–2735.

[17] P. Sundaram, M. Luessi, M. Bianciardi, S. Stufflebeam, M. Hämäläinen, and V. Solo, "Individual resting-state brain networks enabled by massive multivariate conditional mutual information," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1957–1966, Jun. 2020.

[18] B. Zhao, Y. S. Wang, and M. Kolar, "FuDGE: A method to estimate a functional differential graph in a high-dimensional setting," *J. Mach. Learn. Res.*, vol. 23, pp. 1–82, Jan. 2022.

[19] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its Oracle properties," *J. Amer. Stat. Assoc.*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[20] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, Jan. 2008.

[21] H. Zou and R. Li, "One-step sparse estimates in nonconcave penalized likelihood models," *Ann. Statist.*, vol. 36, no. 4, pp. 1509–1533, Aug. 2008.

[22] J. K. Tugnait, "Sparse graph learning under Laplacian-related constraints," *IEEE Access*, vol. 9, pp. 151067–151079, 2021.

[23] J. K. Tugnait, "Sparse-group log-sum penalized graphical model learning for time series," in *Proc. IEEE Intern. Conf. Acoust., Speech Signal Process. (ICASSP)*, Singapore, May 2022, pp. 5822–5826.

[24] Q. Wei and Z. Zhao, "Large covariance matrix estimation with Oracle statistical rate via majorization-minimization," *IEEE Trans. Signal Process.*, vol. 71, pp. 3328–3342, 2023.

[25] R. Varma, H. Lee, J. Kovacevic, and Y. Chi, "Vector-valued graph trend filtering with non-convex penalties," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 6, pp. 48–62, 2020.

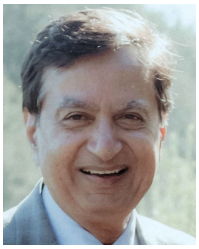
[26] P. Xu and Q. Gu, "Semiparametric differential graph models," in *Proc. 30th Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 29, Barcelona, Spain, 2016, pp. 1–9.

[27] J. K. Tugnait, "Estimation of differential graphs via log-sum penalized D-trace loss," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Hanoi, Vietnam, Jul. 2023, pp. 240–244.

[28] B. Jiang, X. Wang, and C. Leng, "A direct approach for sparse quadratic discriminant analysis," *J. Mach. Learn. Res.*, vol. 19, no. 31, pp. 1–37, Sep. 2018.

[29] T. Zhang and H. Zou, "Sparse precision matrix estimation via lasso penalized D-trace loss," *Biometrika*, vol. 101, no. 1, pp. 103–120, Mar. 2014.

[30] S. Kumar, J. Ying, J. V. d. M. Cardoso, and D. P. Palomar, "Structured graph learning via Laplacian spectral constraints," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, Vancouver, BC, Canada, Sep. 2019, pp. 1–13.
[31] Y. Medvedovsky, E. Treister, and T. Routtenberg, "Efficient graph Laplacian estimation by proximal Newton," in *Proc. 27th Intern. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 238, Valencia, Spain, May 2023, pp. 1171–1179.
[32] D. S. Tracy and K. G. Jinadasa, "Partitioned Kronecker products of matrices and applications," *Can. J. Statist.*, vol. 17, no. 1, pp. 107–120, Mar. 1989.
[33] S. Liu, "Matrix results on the Khatri-Rao and Tracy-Singh products," *Linear Algebra Appl.*, vol. 289, nos. 1–3, pp. 267–277, Mar. 1999.
[34] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 68, no. 1, pp. 49–67, Feb. 2006.
[35] P.-L. Loh and M. J. Wainwright, "Support recovery without incoherence: A case for nonconvex regularization," *Ann. Statist.*, vol. 45, no. 6, pp. 2455–2482, Dec. 2017.
[36] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.
[37] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Mach. Learn.*, vol. 1, no. 3, pp. 127–239, Nov. 2013.
[38] Q. Yao and J. T. Kwok, "Efficient learning with a family of non-convex regularizers by redistributing non-convexity," *J. Mach. Learn. Res.*, vol. 18, pp. 1–52, Jan. 2018.
[39] P. Gong, C. Zhang, Z. Lu, J. Z. Huang, and J. Ye, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," in *Proc. 30th Intern. Conf. Mach. Learn. (ICML)*, 2013, pp. 37–45.
[40] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers," *Stat. Sci.*, vol. 27, no. 4, pp. 538–557, Nov. 2012.
[41] P.-L. Loh and M. J. Wainwright, "Regularized M-estimators with non-convexity: Statistical and algorithmic theory for local optima," *J. Mach. Learn. Res.*, vol. 16, pp. 559–616, 2015.
[42] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA, USA: Athena Scientific, 2003.
[43] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.
[44] S. Lu, J. Kang, W. Gong, and D. Towsley, "Complex network comparison using random walks," in *Proc. 23rd Int. Conf. World Wide Web*, Seoul, (South) Korea, Apr. 2014, pp. 727–730.
[45] S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, and S. X. Chen, "Cautionary tales on air-quality improvement in Beijing," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 473, no. 2205, Sep. 2017, Art. no. 20170457.
[46] W. Chen, F. Wang, G. Xiao, K. Wu, and S. Zhang, "Air quality of Beijing and impacts of the new ambient air quality standard," *Atmosphere*, vol. 6, no. 8, pp. 1243–1258, Aug. 2015.
[47] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.



JITENDRA K. TUGNAIT (Life Fellow, IEEE) received the B.Sc. degree (Hons.) in electronics and electrical communication engineering from Punjab Engineering College, Chandigarh, India, in 1971, the M.S. and E.E. degrees in electrical engineering from Syracuse University, Syracuse, NY, USA, in 1973 and 1974, respectively, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1978.

From 1978 to 1982, he was an Assistant Professor of electrical and computer engineering at the University of Iowa, Iowa City, IA, USA. He was with the Long Range Research Division, Exxon Production Research Company, Houston, TX, USA, from June 1982 to September 1989. In September 1989, he joined the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA, as a Professor, where he is currently James B. Davis Professor. His current research interests include statistical signal processing and machine learning for signal processing.

Dr. Tugnait has served as an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE SIGNAL PROCESSING LETTERS, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; a Senior Area Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING; and a Senior Editor for IEEE WIRELESS COMMUNICATIONS LETTERS.