IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 November 2025, accepted 29 November 2025, date of publication 2 December 2025,
date of current version 8 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3639399

RESEARCH ARTICLE

Learning Conditional Independence Differential Graphs From Time-Dependent Data

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 Estimation of differences in conditional independence graphs (CIGs) of two time series Gaussian graphical models (TSGGMs) is investigated where the two TSGGMs are known to have similar structure. The TSGGM structure is encoded in the inverse power spectral density (IPSD) of the time series. In several existing works, one is interested in estimating the difference in two precision matrices to characterize underlying changes in conditional dependencies of two sets of data consisting of independent and identically distributed (i.i.d.) observations. In this paper we consider estimation of the difference in two IPSDs to characterize the underlying changes in conditional dependencies of two sets of time-dependent data. Our approach accounts for data time dependencies unlike past work. We analyze a penalized D-trace loss function approach in the frequency domain for differential graph learning, using Wirtinger calculus. We consider both convex (group lasso) and non-convex (log-sum and SCAD group penalties) penalty/regularization functions. An alternating direction method of multipliers (ADMM) algorithm is presented to optimize the objective function. We establish sufficient conditions in a high-dimensional setting for consistency (convergence of the inverse power spectral density to true value in the Frobenius norm) and graph recovery. Both synthetic and real data examples are presented in support of the proposed approaches. In synthetic data examples, our log-sum-penalized differential time-series graph estimator significantly outperformed our lasso based differential time-series graph estimator which, in turn, significantly outperformed an existing lasso-penalized i.i.d. modeling approach, with F_1 score as the performance metric. In a 120-dimensional moving-average model based time series example, for sample sizes of $n = 512$ and 4096, our log-sum-penalized estimator improved the F_1 scores by 84% and 44%, respectively, over our lasso-penalized method and by 119% and 112%, respectively, over existing lasso-penalized i.i.d. method (F_1 scores 0.46 and 0.91 for log-sum, 0.28 and 0.58 for proposed lasso, and 0.21 and 0.43 for i.i.d. lasso).

INDEX TERMS Sparse graph learning, differential graphs, time series graphs, non-convex penalties, inverse power spectral density.

I. INTRODUCTION

Graphical models are used to display and explore conditional independence structure in the analysis of multivariate data [1], [2], [3], [4]. Consider a 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 $\mathcal{E} \subseteq [p] \times [p]$. Next consider a stationary (real-valued), zero-mean, $p-$dimensional multivariate Gaussian time series $\boldsymbol{x}(t)$, $t = 0, \pm 1, \pm 2, \cdots$, with ith component $x_i(t)$, and correlation (covariance) matrix function $\boldsymbol{R}_{xx}(\tau) = \mathbb{E}\{\boldsymbol{x}(t+\tau)\boldsymbol{x}^T(t)\}$, $\tau = 0, \pm 1, \cdots$. Given $\{\boldsymbol{x}(t)\}$, in the corresponding graph $\mathcal{G}$, each component series $\{x_i(t)\}$ is represented by a node (i in V), and associations between components $\{x_i(t)\}$ and $\{x_j(t)\}$ are represented by edges between nodes i and j of $\mathcal{G}$. In a conditional independence graph (CIG), there is no edge between nodes i and j (i.e., $\{i, j\} \notin \mathcal{E}$) if and only if (iff) $x_i(t)$ and $x_j(t)$ are conditionally independent given the remaining

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

p-2 scalar series $x_\ell(t)$, $\ell \in [p]$, $\ell \neq i$, $\ell \neq j$. (This is a generalization of CIG for random vectors where $\{i,j\} \notin \mathcal{E}$ iff $\Omega_{ij} = 0$ [1], [2], [3]; $\boldsymbol{\Omega} = (E\{\boldsymbol{x}(t)\boldsymbol{x}^\top(t)\})^{-1}$ is the precision matrix.)

Let $\boldsymbol{S}_x(f)$ denote the power spectral density (PSD) matrix of $\{\boldsymbol{x}(t)\}$, given by $\boldsymbol{S}_x(f) = \sum_{\tau=-\infty}^{\infty} \boldsymbol{R}_{xx}(\tau)e^{-\iota 2\pi f \tau}$, $\iota = \sqrt{-1}$. In [4] it was established that conditional independence of two time series components given all other components of the time series, is encoded by zeros in the inverse PSD (IPSD), that is, $\{i,j\} \notin \mathcal{E}$ iff the (i,j)-th element of $\boldsymbol{S}_x^{-1}(f)$ vanishes, i.e., $[\boldsymbol{S}_x^{-1}(f)]_{ij} = 0$ for every f. Hence one can use estimated IPSD of observed time series to infer the associated graph.

There has also been some interest in differential network analysis where one estimates the difference in two inverse covariance matrices [5], [6], [7], [8]. 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 $[\boldsymbol{\Delta}]_{ij} \neq 0$. It characterizes differences between the Gaussian graphical models (GGMs) of the two sets of data. We use the term differential graph as in [8], [9] ([5], [7], and [10] use the term differential network). As noted in [5] and [7], in biostatistics, the differential network/graph describes the changes in conditional dependencies between components under different environmental or genetic conditions. For instance, one may be interested in the differences in the graphical models of healthy and impaired subjects, or models under different disease states, given gene expression data or functional magnetic resonance imaging (fMRI) signals [11], [12], [13].

In several applications such as fMRI signal analysis or financial time series analysis, the underlying temporal data is not i.i.d. Analysis of some resting state fMRI data in [14] shows significant time-dependence. The data set analyzed in [14, Sec. 2.2] consists of a single subject 1190 temporal brain images, each image with 907 functional brain nodes. This data passed stationarity, Gaussianity and linearity tests [14, Sec. 2.2], implying that Gaussian time series assumption used in this paper would be appropriate. The focus of [14] is analysis of graphical models derived from precision matrix of dependent data; they do not address time series graphical models. In [15, Sec. 5.2] autoregressive models are fitted to financial time series (international stock market data) to infer the underlying time series graphical model. Differential network analysis in such applications calls for consideration of time-dependence in the data as well as consideration of time series graphical models, instead of just assuming that the data is i.i.d., as in [5], [6], [7], [8], and [10]. There is no prior reported work on differential times series graph estimation. This paper attempts to fill this gap.

In this paper we address the problem of estimating differences in two time series Gaussian graphical models (TSGGMs) which are known to have similar structure. Our approach accounts for data time dependencies unlike past work. The TSGGM structure is encoded in its IPSD just as the vector GGM structure is encoded in its precision matrix. We consider estimation of the difference in two IPSD's to characterize underlying changes in conditional dependencies of two sets of time-dependent data $\{\boldsymbol{x}(t)\}_{t=1}^{n_x}$ and $\{\boldsymbol{y}(t)\}_{t=1}^{n_y}$. We analyze a penalized D-trace loss function approach in the frequency domain for differential graph learning, using Wirtinger calculus [16]. As a preliminary step, we first address the problem of estimation of complex differential graphs, given two complex-valued i.i.d. time series. We consider both convex (group lasso) and non-convex (log-sum and Smoothly Clipped Absolute Deviation (SCAD) group penalties) penalty/regularization functions. The use of non-convex penalties (unlike convex lasso penalty) is known to yield more accurate results, i.e., they can produce sparse set of solution like lasso, and approximately unbiased coefficients for large coefficients, unlike lasso [17], [18], [19].

A. RELATED WORK

The problem of estimation of complex differential graphs and the general problem of differential times series graph estimation have not been investigated before. The work of [9] considers time series differential graphs with D-trace loss functions except that in [9] $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ are non-stationary ("functional" modeling), and instead of a single record (sample) of $\boldsymbol{x}(t)$, $t \in [n]$ and $\boldsymbol{y}(t)$, $t \in [n]$, as in this paper, they assume multiple independent observations of $\boldsymbol{x}(t)$, $t \in \mathcal{T}$, and $\boldsymbol{y}(t)$, $t \in \mathcal{T}$ (a closed subset of real line). However, as in [9], we follow the framework of [20] for theoretical analysis. Unlike our paper, [9] does not consider non-convex penalties.

Differential network analysis reported in [5], [6], [7], [8], and [10] all deal with i.i.d. data whereas we address dependent data. In [21] differential latent variable graphical models are estimated assuming i.i.d. data. In latent variable models there are some hidden nodes. We do not consider this aspect in this paper. As in [5], [6], [7], [8], and [10] we use a D-trace loss function approach. One naive approach to differential network analysis with i.i.d. data would be to estimate the two precision matrices separately by any existing estimator (see [3], [19], [22], [23] and references therein) and then calculate their difference to estimate the differential graph. In such an approach one estimates twice the number of parameters (the respective precision matrices instead of the difference), therefore, one needs larger sample sizes for same accuracy. Also, this naive approach imposes sparsity constraints on each precision matrix for the methods to work. The same comment applies to methods such as [11], where the two precision matrices and their differences are jointly estimated. If only the difference in the precision matrices is of interest, direct estimation of the difference in the two precision matrices is preferable and has been considered in [5], [6], [7], [8], [10], and [12], where only the difference is required to be sparse, not the two individual precision matrices. In [5], [6], [7], [8], and [10] precision difference matrix estimators are based on a D-trace loss [24], while [12] discusses a Dantzig selector type estimator. In this paper we extend the D-trace loss framework of [5] and [6] for i.i.d. data to address time-dependent data via a frequency-domain formulation.

Our work exploits prior work on graphical modeling of real time series in high-dimensional settings. Nonparametric approaches for graphical modeling of real time series in high-dimensional settings (p is large and/or sample size n is of the order of p) have been investigated in [25], [26], and [27], among others. We use the frequency-domain formulation of [26] and [27] which deals with graphical modeling of time series, but not with differential graphical modeling addressed here. In [28] and [29] estimation of high-dimensional power spectral matrix is addressed. Time series graphical modeling is not discussed in [29], unlike [28] where testing-based methods are used for inference of graphical models (as opposed to regularization based methods in [25], [26], [27]). Differential graphical modeling is not addressed in [28] and [29].

Although non-convex penalties have been extensively used for graph estimation (see [19], [22], [23] and references therein) and recently for differential graph estimation from i.i.d. data [30], they have not been applied to differential time-series graphs. For optimization we use the standard alternating direction method of multipliers (ADMM) approaches [31] except that our ADMM algorithm applies to real objective function of complex variables exploiting the Wirtinger calculus. Our numerical results show that our log-sum-penalized differential time-series graph estimator significantly outperforms our lasso based differential time-series graph estimator which in turn, significantly outperforms the i.i.d. modeling based time domain methods of [5] and [6] (lasso penalty) and [30] (log-sum penalty), with F_1 score as the performance metric.

Graphical models have also been inferred from consideration other than statistical [32]. One class of graphical models are based on signal smoothness [32], [33], [34] where graph learning from data becomes equivalent to estimation of the graph Laplacian matrix. Some reviews of various graph learning approaches may be found in [35], [36], [37], and [38]. A large variety of graph learning models and approaches exist, motivated by diverse applications in signal processing, machine learning, and other areas. In [37] (also [36]) existing graph learning methods are classified into four broad categories: deep learning based methods, matrix factorization based methods, random walk based methods, and graph signal processing based methods. In terms of these four categories, our approach falls in the category of graph signal processing based methods with the sub-category of "learning topology structure." Differently from [37], [35] categorizes graph learning methods based on two graph construction steps: (1) determine the edge set $\mathcal{E}$, called E-step, and (2) based on $\mathcal{E}$, determine an edge weight matrix $\boldsymbol{W}$, called W-step, even though in some methods these two steps may be merged into one, or the second step may be executed first yielding $\boldsymbol{W}$ which then determines $\mathcal{E}$. In [38] graphical modeling is approached from a statistical viewpoint and a wide variety of models (i.i.d. Gaussian data, matrix-valued data, quantile graphical models, etc.) and approaches are considered. In this paper we are interested in conditional independence differential graphs, a topic not addressed in [35], [36], [37], and [38].

More recently there has been interest in introducing fairness considerations in graphical modeling [39] (based on i.i.d. data assumption), and in exploiting transfer learning ideas in estimating one target graph and several auxiliary graphs using (i.i.d.) data from multiple sources [40]. In [40] concepts similar to differential graphs are used in transferring auxiliary graph structure to the target graph.

A class of graph and graph-based learning approaches are motivated by specific application tasks such as clustering and classification. Examples of such approaches include [41], [42], [43] and relevant references in [35], [36], [37], and [38]. While these approaches address important useful problems, they are not related to the differential time series graph learning problem addressed in this paper. In such approaches an important consideration is how to incorporate prior information relevant to the intended application, in the graph model. For instance, both local and global structure information is incorporated in the model of [41], together with a rank constraint on the graph Laplacian to reflect the number of clusters. In [43] a multi-domain speech emotion recognition problem is addressed where domain discrepancy between target and source domains is captured by similarity and dissimilarity graphs, modeled via Laplacian matrices. Construction of these Laplacian matrices in [43] does not follow any statistical approach or consider conditional independence. That is, the objectives in this paper and [43] are quite different, resulting in distinctly different approaches. In [42] the speech emotion detection problem of [43] is combined with gender prediction, and the focus is on optimization of feature selection. In [42] a particle swarm optimization approach is investigated which is a general heuristic approach which does not guarantee a global optimum. In this paper our penalized D-trace loss function with lasso penalty or local-linear approximated non-convex penalty, is convex and our ADMM optimization algorithm is provably convergent to a global minimum (see Secs. II-C1 and V-A). For such problems ADMM is generally considered to be a computationally efficient optimizer [31].

As noted in [35], "…how to select a suitable graph construction/learning strategy in practice …is a challenging problem without a universal solution, since it depends on many factors …"

B. OUR CONTRIBUTIONS

We first address the problem of estimation of complex differential graphs, given two complex-valued i.i.d. time series. These results form the basis for analyzing a novel penalized D-trace loss function approach in the frequency domain for differential graph learning, using Wirtinger calculus. We consider both convex group lasso and non-convex (log-sum and SCAD) group penalties regularization functions. An ADMM algorithm is presented to optimize the objective function, using a local linear approximation (LLA) [19], [22] based iterative approach for non-convex penalties. Theoretical analysis establishing sufficient conditions for consistency (convergence of the inverse power spectral density to true value in the Frobenius norm) and graph recovery is presented using the framework of [20] which does not apply to the SCAD penalty. Both synthetic and real data examples are presented in support of the proposed approaches.

A preliminary version of this paper appears in a conference paper [44] where non-convex penalties are not considered and no proof is given for [44, Theorem 1] (corresponding to our Theorem 1). Moreover, [44] has no counterparts to our Sec. V-B and Theorem 2, and it has limited synthetic data results and no real data results.

C. NOTATION AND OUTLINE

For a set V, $|V|$ denotes its cardinality. Given $\boldsymbol{A} \in \mathbb{C}^{p\times p}$, we use $\phi_{\min}(\boldsymbol{A})$, $\phi_{\max}(\boldsymbol{A})$, $|\boldsymbol{A}|$ and $\mathrm{tr}(\boldsymbol{A})$ to denote the minimum eigenvalue, maximum eigenvalue, determinant and trace of $\boldsymbol{A}$, respectively, and we use $\boldsymbol{A} \succ \boldsymbol{0}$ and $\boldsymbol{A} \succeq \boldsymbol{0}$ to denote that $\boldsymbol{A}$ is positive-definite and positive semi-definite, respectively. Given $\boldsymbol{B} \in \mathbb{C}^{p\times m}$, $[\boldsymbol{B}]_{ij}$ denotes the (i,j)-th element of $\boldsymbol{B}$, and so does B_{ij}, and $\boldsymbol{I}_q$ denotes the $q \times q$ identity matrix. The symbol $\otimes$ denotes the matrix Kronecker product and the symbol $\circ$ denotes the Hadamard product. The superscripts $*$ and H denote the complex conjugate and the Hermitian (conjugate transpose) operations, respectively.

For $\boldsymbol{B} \in \mathbb{C}^{p\times q}$, we define the operator norm, the Frobenius norm and the vectorized ℓ_1 norm, respectively, as $\|\boldsymbol{B}\| = \sqrt{\phi_{\max}(\boldsymbol{B}^H\boldsymbol{B})}$, $\|\boldsymbol{B}\|_F = \sqrt{\mathrm{tr}(\boldsymbol{B}^H\boldsymbol{B})}$, $\|\boldsymbol{B}\|_1 = \sum_{i,j}|B_{ij}|$ and $\|\boldsymbol{B}\|_\infty = \max_{i,j}|B_{ij}|$. For vector $\boldsymbol{\theta} \in \mathbb{C}^p$, we define $\|\boldsymbol{\theta}\|_1 = \sum_{i=1}^p|\theta_i|$ and $\|\boldsymbol{\theta}\|_2 = \sqrt{\sum_{i=1}^p|\theta_i|^2}$, and we also use $\|\boldsymbol{\theta}\|$ for $\|\boldsymbol{\theta}\|_2$. Given $\boldsymbol{A} \in \mathbb{C}^{n\times p}$, column vector $\mathrm{vec}(\boldsymbol{A}) \in \mathbb{C}^{np}$ denotes the vectorization of $\boldsymbol{A}$ which stacks the columns of the matrix $\boldsymbol{A}$, and $\mathrm{Re}(\boldsymbol{A})$ and $\mathrm{Im}(\boldsymbol{A})$ denote the real and imaginary parts, respectively, of $\boldsymbol{A}$. The notation $\boldsymbol{x} \sim \mathcal{N}_c(\boldsymbol{m}, \boldsymbol{\Sigma})$ denotes a random vector $\boldsymbol{x}$ that is circularly symmetric (proper) complex Gaussian with mean $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$. Similarly, $\boldsymbol{x} \sim \mathcal{N}_r(\boldsymbol{m}, \boldsymbol{\Sigma})$ denotes a random vector $\boldsymbol{x}$ that is real-valued Gaussian with mean $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$. Given a variable vector $\boldsymbol{x}$ or matrix $\boldsymbol{X}$, we use $\boldsymbol{x}^\diamond$ or $\boldsymbol{X}^\diamond$, respectively, to denote their true values.

The rest of the paper is organized as follows. A penalized D-trace loss function is presented in Sec. II for estimation of complex differential graphs, given two complex-valued i.i.d. time series. These results form the basis for a novel penalized D-trace loss function approach in the frequency domain for differential graph learning, formulated in Secs. III and IV. A solution to optimization of the penalized D-trace loss is provided in Sec. V and the selection of the tuning parameters is presented in Sec. V-B. In Sec. VI we provide a theoretical analysis of the proposed approach, resulting in Theorems 1 and 2. Numerical results are presented in Secs. VII and VIII. A derivation of (32) and the proofs of Theorems 1 and 2 are given in the two appendices.

II. COMPLEX DIFFERENTIAL GRAPHS

As a preliminary step, we first address the problem of estimation of sparse complex differential graphs, given two complex-valued i.i.d. time series. To this end, we first review the problem of estimation of real differential graphs in Sec. II-A. The results of Sec. II-A are then extended to complex differential graphs in Secs. II-B and II-C. The results of Secs. II-B and II-C are later exploited in Secs. III, IV and V to address time series differential graphs.

A. REAL GAUSSIAN VECTORS

We first recall a formulation of [5], [6], [7], and [10] for real-valued data. Let $\boldsymbol{x} \in \mathbb{R}^p$, $\boldsymbol{x} \sim \mathcal{N}_r(\boldsymbol{0}, \boldsymbol{\Sigma}_x^\diamond)$, $\boldsymbol{\Sigma}_x^\diamond \succ \boldsymbol{0}$, and suppose we are given i.i.d. samples $\{\boldsymbol{x}(t)\}_{t=1}^{n_x}$ of $\boldsymbol{x}$, and similarly given i.i.d. samples $\{\boldsymbol{y}(t)\}_{t=1}^{n_y}$ of independent $\boldsymbol{y} \in \mathbb{R}^p$, $\boldsymbol{y} \sim \mathcal{N}_r(\boldsymbol{0}, \boldsymbol{\Sigma}_y^\diamond)$, $\boldsymbol{\Sigma}_y^\diamond \succ \boldsymbol{0}$. Let $\boldsymbol{\Omega}_y^\diamond = (\boldsymbol{\Sigma}_y^\diamond)^{-1}$ and $\boldsymbol{\Omega}_x^\diamond = (\boldsymbol{\Sigma}_x^\diamond)^{-1}$ denote the respective precision matrices, and let $\hat{\boldsymbol{\Sigma}}_x = \frac{1}{n_x}\sum_{t=1}^{n_x}\boldsymbol{x}(t)\boldsymbol{x}^\top(t)$ and $\hat{\boldsymbol{\Sigma}}_y = \frac{1}{n_y}\sum_{t=1}^{n_y}\boldsymbol{y}(t)\boldsymbol{y}^\top(t)$ denote the sample covariance estimates. In [5], [6], [7], and [10] one seeks to estimate $\boldsymbol{\Delta}^\diamond = \boldsymbol{\Omega}_y^\diamond - \boldsymbol{\Omega}_x^\diamond$ and graph $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$, based on $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$.

In [5] (see also [6, Sec. 2.1]), the following convex D-trace loss function is used for $\boldsymbol{\Delta} \in \mathbb{R}^{p\times p}$

$$L_r(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) = \frac{1}{2}\mathrm{tr}(\hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y\boldsymbol{\Delta}^\top) - \mathrm{tr}(\boldsymbol{\Delta}(\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y)) \quad (1)$$

where D-trace refers to difference-in-trace loss function, a term coined in [24] in the context of graphical model estimation.

Using the rule [45, p. 13,(117)] regarding matrix differentiation of a trace function, we have

$$\frac{\partial\,\mathrm{tr}(\hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y\boldsymbol{\Delta}^\top)}{\partial\boldsymbol{\Delta}} = 2\hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y \quad (2)$$

and using [45, p. 12,(100)], we have

$$\frac{\partial\,\mathrm{tr}(\boldsymbol{\Delta}(\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y))}{\partial\boldsymbol{\Delta}} = \hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y\,. \quad (3)$$

It then follows that

$$\frac{\partial L_r(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)}{\partial\boldsymbol{\Delta}} = \hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y - (\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y) \quad (4)$$

and therefore, at the true covariances, we have

$$\frac{\partial L_r(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)}{\partial\boldsymbol{\Delta}} = \boldsymbol{\Sigma}_x^\diamond\boldsymbol{\Delta}\boldsymbol{\Sigma}_y^\diamond - (\boldsymbol{\Sigma}_x^\diamond - \boldsymbol{\Sigma}_y^\diamond)\,. \quad (5)$$

When $\boldsymbol{\Delta} = \boldsymbol{\Delta}^\diamond = \boldsymbol{\Omega}_y^\diamond - \boldsymbol{\Omega}_x^\diamond$, we have

$$\begin{aligned}\boldsymbol{\Sigma}_x^\diamond\boldsymbol{\Delta}\boldsymbol{\Sigma}_y^\diamond &= \boldsymbol{\Sigma}_x^\diamond(\boldsymbol{\Omega}_y^\diamond - \boldsymbol{\Omega}_x^\diamond)\boldsymbol{\Sigma}_y^\diamond \\ &= \boldsymbol{\Sigma}_x^\diamond\boldsymbol{\Omega}_y^\diamond\boldsymbol{\Sigma}_y^\diamond - \boldsymbol{\Sigma}_x^\diamond\boldsymbol{\Omega}_x^\diamond\boldsymbol{\Sigma}_y^\diamond = \boldsymbol{\Sigma}_x^\diamond - \boldsymbol{\Sigma}_y^\diamond\,, \end{aligned} \quad (6)$$

and therefore, by (5),

$$\frac{\partial L_r(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)}{\partial \boldsymbol{\Delta}}\Big|_{\boldsymbol{\Delta}=\boldsymbol{\Delta}^\diamond} = \mathbf{0}\,. \tag{7}$$

Using $\text{tr}(\boldsymbol{A}^\top \boldsymbol{B}\boldsymbol{C}\boldsymbol{D}^\top) = \text{vec}(\boldsymbol{A})^\top(\boldsymbol{D}\otimes\boldsymbol{B})\text{vec}(\boldsymbol{C})$ and letting $\boldsymbol{d} := \text{vec}(\boldsymbol{\Delta}) \in \mathbb{R}^{p^2}$, we have

$$L_r(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond) = \frac{1}{2}\boldsymbol{d}^\top(\boldsymbol{\Sigma}_y^\diamond \otimes \boldsymbol{\Sigma}_x^\diamond)\boldsymbol{d} - \boldsymbol{d}^\top \text{vec}(\boldsymbol{\Sigma}_x^\diamond - \boldsymbol{\Sigma}_y^\diamond)\,. \tag{8}$$

Thus $L_r(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)$ is quadratic in $\boldsymbol{d}$ with the Hessian matrix given by

$$\boldsymbol{H}_r = \boldsymbol{\Sigma}_y^\diamond \otimes \boldsymbol{\Sigma}_x^\diamond\,, \quad [\boldsymbol{H}_r]_{k\ell} = \frac{\partial^2 L_r(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)}{\partial[\boldsymbol{d}]_k \partial[\boldsymbol{d}]_\ell}\,. \tag{9}$$

The eigenvalues of $\boldsymbol{H}_r$ are the product of the eigenvalues of $\boldsymbol{\Sigma}_y^\diamond$ and $\boldsymbol{\Sigma}_x^\diamond$. By assumption $\boldsymbol{\Sigma}_y^\diamond$ and $\boldsymbol{\Sigma}_x^\diamond$ are positive-definite, hence, $\boldsymbol{H}_r \succ \mathbf{0}$ since all eigenvalues of Hermitian $\boldsymbol{H}_r$ are positive. Therefore, the function $L_r(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)$ is strictly convex in $\boldsymbol{\Delta}$ and by (7), has a unique minimum at $\boldsymbol{\Delta}^\diamond = \boldsymbol{\Omega}_y^\diamond - \boldsymbol{\Omega}_x^\diamond$ [5], [6].

Since the true data covariances are unavailable, one uses sample covariances of the data and $\boldsymbol{\Delta}$ is estimated by minimizing a lasso-penalized D-trace loss function (1) [5], [6], [7], [10], given by $L_r(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) + \lambda \sum_{k,\ell=1}^p |\Delta_{k\ell}|$. With $V = [p]$ and $\mathcal{E} \subseteq [p] \times [p]$ the associated differential graph is $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$ where $\{i, j\} \in \mathcal{E}_\Delta$ iff $[\boldsymbol{\Delta}]_{ij} \neq 0$.

B. PROPER COMPLEX GAUSSIAN VECTORS

Consider complex-valued $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^p$, with $\boldsymbol{x} \sim \mathcal{N}_c(\mathbf{0}, \boldsymbol{\Sigma}_x^\diamond)$ and $\boldsymbol{y} \sim \mathcal{N}_c(\mathbf{0}, \boldsymbol{\Sigma}_y^\diamond)$ with $\boldsymbol{\Sigma}_x^\diamond \succ \mathbf{0}$ and $\boldsymbol{\Sigma}_y^\diamond \succ \mathbf{0}$. Given i.i.d. measurements $\{\boldsymbol{x}(t)\}_{t=1}^{n_x}$ and $\{\boldsymbol{y}(t)\}_{t=1}^{n_y}$, we desire to estimate $\boldsymbol{\Delta}^\diamond = \boldsymbol{\Omega}_y^\diamond - \boldsymbol{\Omega}_x^\diamond$.

We propose a real-valued cost of complex-valued $\boldsymbol{\Delta} \in \mathbb{C}^{p\times p}$ as

$$L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) = \frac{1}{2}\Big(\text{tr}(\hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y\boldsymbol{\Delta}^H) + \text{tr}(\hat{\boldsymbol{\Sigma}}_x^*\boldsymbol{\Delta}^*\hat{\boldsymbol{\Sigma}}_y^*\boldsymbol{\Delta}^\top)\Big) - \text{tr}\Big(\boldsymbol{\Delta}(\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y) + \boldsymbol{\Delta}^*(\hat{\boldsymbol{\Sigma}}_x^* - \hat{\boldsymbol{\Sigma}}_y^*)\Big) \tag{10}$$

with $\hat{\boldsymbol{\Sigma}}_x = \frac{1}{n_x}\sum_{t=1}^{n_x} \boldsymbol{x}(t)\boldsymbol{x}^H(t)$ and $\hat{\boldsymbol{\Sigma}}_y = \frac{1}{n_y}\sum_{t=1}^{n_y} \boldsymbol{y}(t)\boldsymbol{y}^H(t)$.

We will use Wirtinger calculus [16, Appendix 2] to analyze $L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)$ where we view it as a real-valued function of two "independent" complex-valued vectors $\text{vec}(\boldsymbol{\Delta})$ and its conjugate $\text{vec}(\boldsymbol{\Delta}^*)$. Similar to (2) but using [45, p. 12,(100),(103)] and the fact that $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$ are Hermitian, we have

$$\frac{\partial \text{tr}(\hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y\boldsymbol{\Delta}^H)}{\partial \boldsymbol{\Delta}^*} = \hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y = \frac{\partial \text{tr}(\hat{\boldsymbol{\Sigma}}_x^*\boldsymbol{\Delta}^*\hat{\boldsymbol{\Sigma}}_y^*\boldsymbol{\Delta}^\top)}{\partial \boldsymbol{\Delta}^*} \tag{11}$$

and similar to (3), we have

$$\frac{\partial \text{tr}(\boldsymbol{\Delta}^*(\hat{\boldsymbol{\Sigma}}_x^* - \hat{\boldsymbol{\Sigma}}_y^*))}{\partial \boldsymbol{\Delta}^*} = \hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y\,. \tag{12}$$

Note that $\frac{\partial \text{tr}(\boldsymbol{\Delta}(\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y))}{\partial \boldsymbol{\Delta}^*} = \mathbf{0}$. It then follows that

$$\frac{\partial L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)}{\partial \boldsymbol{\Delta}^*} = \hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}\hat{\boldsymbol{\Sigma}}_y - (\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y) \tag{13}$$

and therefore, at the true covariances, we have

$$\frac{\partial L(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)}{\partial \boldsymbol{\Delta}^*} = \boldsymbol{\Sigma}_x^\diamond\boldsymbol{\Delta}\boldsymbol{\Sigma}_y^\diamond - (\boldsymbol{\Sigma}_x^\diamond - \boldsymbol{\Sigma}_y^\diamond)\,. \tag{14}$$

When $\boldsymbol{\Delta} = \boldsymbol{\Delta}^\diamond = \boldsymbol{\Omega}_y^\diamond - \boldsymbol{\Omega}_x^\diamond$, by (6), we have

$$\frac{\partial L(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)}{\partial \boldsymbol{\Delta}^*}\Big|_{\boldsymbol{\Delta}=\boldsymbol{\Delta}^\diamond} = \mathbf{0}\,. \tag{15}$$

Define

$$\boldsymbol{\theta} = [\text{vec}(\boldsymbol{\Delta})^\top \ \text{vec}(\boldsymbol{\Delta})^H]^\top \in \mathbb{C}^{2p^2}\,, \tag{16}$$

$$\boldsymbol{H} = \begin{bmatrix} (\boldsymbol{\Sigma}_y^\diamond)^* \otimes \boldsymbol{\Sigma}_x^\diamond & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_y^\diamond \otimes (\boldsymbol{\Sigma}_x^\diamond)^* \end{bmatrix}, \tag{17}$$

$$\boldsymbol{b} = \begin{bmatrix} \text{vec}(\boldsymbol{\Sigma}_x^\diamond - \boldsymbol{\Sigma}_y^\diamond) \\ \text{vec}((\boldsymbol{\Sigma}_x^\diamond)^* - (\boldsymbol{\Sigma}_y^\diamond)^*) \end{bmatrix}. \tag{18}$$

Using $\text{tr}(\boldsymbol{A}^\top \boldsymbol{B}\boldsymbol{C}\boldsymbol{D}^\top) = \text{vec}(\boldsymbol{A})^\top(\boldsymbol{D}\otimes\boldsymbol{B})\text{vec}(\boldsymbol{C})$, we have

$$L(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond) = \frac{1}{2}\boldsymbol{\theta}^H\boldsymbol{H}\boldsymbol{\theta} - \boldsymbol{\theta}^H\boldsymbol{b}\,. \tag{19}$$

Clearly $L(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)$ is quadratic in $\boldsymbol{\theta}$ with the complex augmented Hessian matrix [16, Def. A2.5] $\boldsymbol{H}$ given by (17), satisfying

$$[\boldsymbol{H}]_{k\ell} = \frac{\partial^2 L(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)}{\partial[\boldsymbol{\theta}]_k \partial[\boldsymbol{\theta}]_\ell^*}\,. \tag{20}$$

The eigenvalues of $\boldsymbol{H}$ are the product of the eigenvalues of $\boldsymbol{\Sigma}_y^\diamond$ and $\boldsymbol{\Sigma}_x^\diamond$. By assumption $\boldsymbol{\Sigma}_y^\diamond$ and $\boldsymbol{\Sigma}_x^\diamond$ are Hermitian positive-definite, hence, $\boldsymbol{H} \succ \mathbf{0}$ since all eigenvalues of Hermitian $\boldsymbol{H}$ are positive. Therefore, the function $L(\boldsymbol{\Delta}, \boldsymbol{\Sigma}_x^\diamond, \boldsymbol{\Sigma}_y^\diamond)$ is strictly convex in $\boldsymbol{\Delta}$ and by (15), has a unique minimum at $\boldsymbol{\Delta}^\diamond = \boldsymbol{\Omega}_y^\diamond - \boldsymbol{\Omega}_x^\diamond$.

Since the true data covariances are unavailable, one uses sample covariances of the data as in (10). In this case we replace $\boldsymbol{\Sigma}_x^\diamond$ and $\boldsymbol{\Sigma}_y^\diamond$ in (17) with $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$, respectively, resulting in the Hessian matrix $\hat{\boldsymbol{H}}$

$$\hat{\boldsymbol{H}} = \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_y^* \otimes \hat{\boldsymbol{\Sigma}}_x & \mathbf{0} \\ \mathbf{0} & \hat{\boldsymbol{\Sigma}}_y \otimes \hat{\boldsymbol{\Sigma}}_x^* \end{bmatrix}. \tag{21}$$

Since $\hat{\boldsymbol{\Sigma}}_x \succeq \mathbf{0}$ and $\hat{\boldsymbol{\Sigma}}_y \succeq \mathbf{0}$, we now have $\hat{\boldsymbol{H}} \succeq \mathbf{0}$, implying that $L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)$ is convex, but not necessarily strictly convex. In the high-dimensional case of $\min\{n_x, n_y\}$ less than or of the order of p, to enforce sparsity and to make the problem well-conditioned, for $\lambda > 0$, define the lasso-penalized D-trace loss (similar to [5], [6], [7], [10])

$$L_\lambda(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) = L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) + \lambda \sum_{k,\ell=1}^p |\Delta_{k\ell}|\,. \tag{22}$$

We seek $\hat{\boldsymbol{\Delta}} = \arg\min_{\boldsymbol{\Delta}} L_\lambda(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)$. We discuss this aspect next in Sec. II-C. With $V = [p]$ and $\mathcal{E} \subseteq [p] \times [p]$, the associated complex differential graph is $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$ where $\{i, j\} \in \mathcal{E}_\Delta$ iff $|[\boldsymbol{\Delta}]_{ij}| > 0$. Even though $\boldsymbol{\Delta}$ is Hermitian, $\hat{\boldsymbol{\Delta}}$ is not necessarily so. We make it Hermitian by setting $\hat{\boldsymbol{\Delta}}_H = \frac{1}{2}(\hat{\boldsymbol{\Delta}} + \hat{\boldsymbol{\Delta}}^H)$, after obtaining $\hat{\boldsymbol{\Delta}}$.

C. OPTIMIZATION

Similar to [6] (also [5], [8]), we use an alternating direction method of multipliers (ADMM) approach [31] with variable splitting to compute $\hat{\boldsymbol{\Delta}} = \arg\min_{\boldsymbol{\Delta}} L_\lambda(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)$. Using variable splitting and adding the equality constraint $\boldsymbol{W} = \boldsymbol{\Delta}$, consider

$$\min_{\boldsymbol{\Delta}, \boldsymbol{W}} \left\{ L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) + \lambda \sum_{k,\ell=1}^{p} |W_{k\ell}| \right\}, \tag{23}$$

$$\text{subject to } \boldsymbol{\Delta} = \boldsymbol{W}, \tag{24}$$

where in the penalty we use $\boldsymbol{W}$ instead of $\boldsymbol{\Delta}$. Let $\boldsymbol{U} \in \mathbb{C}^{p\times p}$ denote the dual variable and $\rho > 0$ denote the penalty parameter in the ADMM algorithm. The scaled augmented Lagrangian for this problem is [31]

$$L_\rho(\boldsymbol{\Delta}, \boldsymbol{W}, \boldsymbol{U}) := L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) \tag{25}$$

$$+ \lambda \sum_{k,\ell=1}^{p} |W_{k\ell}| + \frac{\rho}{2}\|\boldsymbol{\Delta} - \boldsymbol{W} + \boldsymbol{U}\|_F^2 . \tag{26}$$

Given the ith iteration results $\boldsymbol{\Delta}^{(i)}, \boldsymbol{W}^{(i)}, \boldsymbol{U}^{(i)}$, in the $(i+1)$st iteration, the ADMM algorithm executes the following 3 updates [31]:

(a) $\boldsymbol{\Delta}^{(i+1)} \leftarrow \arg\min_{\boldsymbol{\Delta}} \bar{L}_a(\boldsymbol{\Delta})$ where

$$\bar{L}_a(\boldsymbol{\Delta}) := L(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y) + \frac{\rho}{2}\|\boldsymbol{\Delta} - \boldsymbol{W}^{(i)} + \boldsymbol{U}^{(i)}\|_F^2 .$$

(b) $\boldsymbol{W}^{(i+1)} \leftarrow \arg\min_{\boldsymbol{W}} \bar{L}_b(\boldsymbol{W})$ where

$$\bar{L}_b(\boldsymbol{W}) := \lambda \sum_{k,\ell=1}^{p} |W_{k\ell}| + \frac{\rho}{2}\|\boldsymbol{\Delta}^{(i+1)} - \boldsymbol{W} + \boldsymbol{U}^{(i)}\|_F^2 .$$

(c) $\boldsymbol{U}^{(i+1)} \leftarrow \boldsymbol{U}^{(i)} + \left(\boldsymbol{\Delta}^{(i+1)} - \boldsymbol{W}^{(i+1)}\right)$

Observe that $\bar{L}_a(\boldsymbol{\Delta})$ and $\bar{L}_b(\boldsymbol{W})$ are convex in $\boldsymbol{\Delta}$ and $\boldsymbol{W}$, respectively.

We now address updates (a) and (b).

Update (a). To minimize $\bar{L}_a(\boldsymbol{\Delta})$ w.r.t. $\boldsymbol{\Delta}$, using Wirtinger calculus, we set

$$\boldsymbol{0} = \frac{\partial \bar{L}_a(\boldsymbol{\Delta})}{\partial \boldsymbol{\Delta}^*}$$
$$= \hat{\boldsymbol{\Sigma}}_x \boldsymbol{\Delta} \hat{\boldsymbol{\Sigma}}_y - (\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y) + \frac{\rho}{2}(\boldsymbol{\Delta} - \boldsymbol{W}^{(i)} + \boldsymbol{U}^{(i)}). \tag{27}$$

Using $\text{vec}(\boldsymbol{AYB}) = (\boldsymbol{B}^\top \otimes \boldsymbol{A})\text{vec}(\boldsymbol{Y})$, we vectorize (27) to obtain

$$\big(\hat{\boldsymbol{\Sigma}}_y^* \otimes \hat{\boldsymbol{\Sigma}}_x + \frac{\rho}{2}\boldsymbol{I}_p \otimes \boldsymbol{I}_p\big)vec(\boldsymbol{\Delta})$$
$$= \text{vec}\big(\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y + \frac{\rho}{2}(\boldsymbol{W}^{(i)} - \boldsymbol{U}^{(i)})\big). \tag{28}$$

Direct matrix inversion solution of (28) requires inversion of a $p^2 \times p^2$ matrix. A computationally cheaper solution follows similar to that in [5] and [6] where the real-valued case is addressed and here we consider complex-valued Hermitian matrices. Carry out eigen-decomposition of $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$ as

$$\hat{\boldsymbol{\Sigma}}_x = \boldsymbol{Q}_x \boldsymbol{D}_x \boldsymbol{Q}_x^H, \quad \boldsymbol{Q}_x \boldsymbol{Q}_x^H = \boldsymbol{I}_p, \tag{29}$$

$$\hat{\boldsymbol{\Sigma}}_y = \boldsymbol{Q}_y \boldsymbol{D}_y \boldsymbol{Q}_y^H, \quad \boldsymbol{Q}_y \boldsymbol{Q}_y^H = \boldsymbol{I}_p, \tag{30}$$

where $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ are diagonal matrices. Define a matrix $\boldsymbol{D} \in \mathbb{R}^{p\times p}$ that organizes the diagonal of $(\boldsymbol{D}_y \otimes \boldsymbol{D}_x + \frac{\rho}{2}\boldsymbol{I}_{p^2})^{-1}$ in a matrix with (j,k)th element as

$$[\boldsymbol{D}]_{jk} = \frac{1}{[\boldsymbol{D}_x]_{jj}[\boldsymbol{D}_y]_{kk} + \frac{\rho}{2}}, \tag{31}$$

and recall that the symbol $\circ$ denotes the Hadamard matrix product. Then $\hat{\boldsymbol{\Delta}}$ that minimizes $\bar{L}_a(\boldsymbol{\Delta})$ is given by (the derivation of (32) is given in Appendix A)

$$\hat{\boldsymbol{\Delta}} = \boldsymbol{Q}_x \Big[\boldsymbol{D} \circ [\boldsymbol{Q}_x^H (\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y + \frac{\rho}{2}(\boldsymbol{W}^{(i)} - \boldsymbol{U}^{(i)}))\boldsymbol{Q}_y]\Big]\boldsymbol{Q}_y^H \tag{32}$$

Note that the eigen-decomposition of $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$ has to be done only once. With $\boldsymbol{A}^{(i)} = \boldsymbol{W}^{(i)} - \boldsymbol{U}^{(i)}$, we have

$$\boldsymbol{\Delta}^{(i+1)} = \boldsymbol{Q}_x \Big[\boldsymbol{D} \circ [\boldsymbol{Q}_x^H (\hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y + \frac{\rho}{2}\boldsymbol{A}^{(i)})\boldsymbol{Q}_y]\Big]\boldsymbol{Q}_y^H . \tag{33}$$

Update (b). Notice that $\bar{L}_b(\boldsymbol{W})$ is separable in (k,ℓ) with

$$\bar{L}_b(W_{k\ell}) = \lambda|W_{k\ell}| + \frac{\rho}{2}|\Delta_{k\ell}^{(i+1)} - W_{k\ell} + U_{k\ell}^{(i)}|^2, \tag{34}$$

$$\bar{L}_b(\boldsymbol{W}) = \sum_{k,\ell=1}^{p} \bar{L}_b(W_{k\ell}). \tag{35}$$

Following [26, Lemma 1] and using $(a)_+ = \max(0,a)$, $\bar{L}_b(W_{k\ell})$ is minimized by the lasso solution

$$W_{k\ell}^{(i+1)} = \left(1 - \frac{(\lambda/\rho)}{|[\boldsymbol{\Delta}^{(i+1)} + \boldsymbol{U}^{(i)}]_{k\ell}|}\right)_+ [\boldsymbol{\Delta}^{(i+1)} + \boldsymbol{U}^{(i)}]_{k\ell} . \tag{36}$$

1) CONVERGENCE

A stopping (convergence) criterion following [31, Sec. 3.3.1] can be devised. The stopping criterion is based on primal and dual residuals being small where, in our case, at $(i+1)$st iteration, the primal residual is given by $\boldsymbol{\Delta}^{(i+1)} - \boldsymbol{W}^{(i+1)}$ and the dual residual by $\rho(\boldsymbol{W}^{(i+1)} - \boldsymbol{W}^{(i)})$. The convergence criterion is met when the norms of these residuals are below some threshold.

The objective function $L_\lambda(\boldsymbol{\Delta}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\Sigma}}_y)$, given by (22), is convex. It is also closed, proper and lower semi-continuous. Hence, for any fixed $\rho > 0$, the ADMM algorithm is guaranteed to converge [31, Sec. 3.2 and Appendix A], in the sense that we have primal residual convergence to 0, dual residual convergence to 0, and the objective function convergence to the optimal value.

III. DIFFERENTIAL TIME SERIES GRAPHS: SYSTEM MODEL

We now turn to the general problem of differential times series graph estimation. Consider two independent stationary (real-valued), zero-mean, $p-$dimensional multivariate Gaussian time series $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$, $t \in \mathbb{Z}$, with PSDs $\boldsymbol{S}_x(f) \succ \boldsymbol{0}$ and $\boldsymbol{S}_y(f) \succ \boldsymbol{0}$, respectively, for every $f \in [0,1]$, and the CIGs $\mathcal{G}_x = (V, \mathcal{E}_x)$ and $\mathcal{G}_y = \big(V, \mathcal{E}_y\big)$, respectively. As discussed earlier in Sec. I, the edge $\{i,j\} \notin \mathcal{E}_x$ iff $[\boldsymbol{S}_x^{-1}(f)]_{ij} \equiv 0$ and $\{i,j\} \notin \mathcal{E}_y$ iff $[\boldsymbol{S}_y^{-1}(f)]_{ij} \equiv 0$ [4]. In differential network analysis for time-dependent data, one is interested in the difference $\boldsymbol{\Delta}(f) = \boldsymbol{S}_y^{-1}(f) - \boldsymbol{S}_x^{-1}(f)$, and in the associated differential graph $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$ we have $\{i,j\} \notin \mathcal{E}_\Delta$ iff $[\boldsymbol{\Delta}(f)]_{ij} = 0$ for every $f \in [0,1]$.

Given data $\{\boldsymbol{x}(t)\}$ and $\{\boldsymbol{y}(t)\}$, our objective is to first estimate the inverse PSDs $\boldsymbol{S}_x^{-1}(f)$ and $\boldsymbol{S}_y^{-1}(f)$ at distinct frequencies, and then select the edge $\{i,j\}$ in the differential time series graph $\mathcal{G}_\Delta$ based on whether or not $[\boldsymbol{\Delta}(f)]_{ij} = 0$ for every $f \in [0,1]$.

A. PROBLEM FORMULATION

Given time-domain data $\{\boldsymbol{x}(t)\}_{t=1}^{n_x}$ and $\{\boldsymbol{y}(t)\}_{t=1}^{n_y}$ originating from two independent stationary, zero-mean, multivariate Gaussian time series $\boldsymbol{x}(t) \in \mathbb{R}^p$ and $\boldsymbol{y}(t) \in \mathbb{R}^p$. For simplicity, we take $n_x = n_y = n$. With $\iota = \sqrt{-1}$, define the (normalized) discrete Fourier transforms (DFTs)

$$\boldsymbol{d}_x(f_m) = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} \boldsymbol{x}(t) \exp\left(-\iota 2\pi f_m (t-1)\right), \tag{37}$$

$$\boldsymbol{d}_y(f_m) = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} \boldsymbol{y}(t) \exp\left(-\iota 2\pi f_m (t-1)\right), \tag{38}$$

$$f_m = \frac{m}{n}, \quad m = 0, 1, \cdots, n-1. \tag{39}$$

The set of complex random vectors $\{\boldsymbol{d}_x(f_m), \boldsymbol{d}_y(f_m)\}_{m=0}^{n/2}$ is a sufficient statistic for any inference problem based on data set $\{\boldsymbol{x}(t), \boldsymbol{y}(t)\}_{t=1}^{n}$ since given $\{\boldsymbol{d}_x(f_m), \boldsymbol{d}_y(f_m)\}_{m=0}^{n/2}$, one can recover $\{\boldsymbol{x}(t), \boldsymbol{y}(t)\}_{t=1}^{n}$ via inverse DFT. [26].

1) MODEL ASSUMPTIONS

We assume the following:

(A1) The time series $\{\boldsymbol{x}(t)\}_{t \in \mathbb{Z}}$ is zero-mean stationary and Gaussian, satisfying

$$\sum_{\tau=-\infty}^{\infty} |[\boldsymbol{R}_{xx}(\tau)]_{k\ell}| < \infty \text{ for every } k, \ell \in V = [p],$$

and similarly for $\{\boldsymbol{y}(t)\}_{t \in \mathbb{Z}}$.

(A2) For some integer $m_t > 0$, let $K = 2m_t + 1$. Pick

$$M = \left\lfloor \left(\frac{n}{2} - m_t - 1\right)/K \right\rfloor,$$
$$\tilde{f}_k = ((k-1)K + m_t + 1)/n \quad \text{for} \quad k \in [M],$$

yielding M equally spaced frequencies $\tilde{f}_k$ in the interval $(0, 0.5)$. Assume that for $\ell = -m_t, -m_t+1, \cdots, m_t$, the PSD matrices $\boldsymbol{S}_x(f)$ and $\boldsymbol{S}_y(f)$ satisfy

$$\boldsymbol{S}_x(\tilde{f}_{k,\ell}) = \boldsymbol{S}_x(\tilde{f}_k), \quad \boldsymbol{S}_y(\tilde{f}_{k,\ell}) = \boldsymbol{S}_y(\tilde{f}_k), \tag{40}$$

$$\text{where } \tilde{f}_{k,\ell} = \big((k-1)K + m_t + 1 + \ell\big)/n. \tag{41}$$

Assumption (A1) is needed to invoke [46, Theorem 4.4.1] regarding distribution of the DFTs $\{\boldsymbol{d}_x(f_m), \boldsymbol{d}_y(f_m)\}_{m=0}^{n/2}$. Assumption (A2) is a local smoothness assumption which implies that $\boldsymbol{S}_x(f_k)$ and $\boldsymbol{S}_y(f_k)$ are constant over $K = 2m_t + 1$ consecutive frequency points f_m's, $m_t > 0$.

It turns out that for "large" n, under assumption (A1), the DFT $\boldsymbol{d}_x(f_m)$ is a complex-valued proper (i.e., circularly symmetric) Gaussian vector $\sim \mathcal{N}_c(\boldsymbol{0}, \boldsymbol{S}_x(f_m))$, and (mutually) independent for $m = 1, 2, \cdots, (n/2) - 1$, ($n$ even) [46, Theorem 4.4.1], though not identically distributed. Also, $\boldsymbol{d}_x(f_0)$ and $\boldsymbol{d}_x(f_{n/2})$ (n even) are real-valued Gaussian vectors. Similar comments apply to $\boldsymbol{d}_y(f_m)$. We exclude the frequency points f_0 and $f_{n/2}$ from further consideration. Define

$$\hat{\boldsymbol{S}}_{xk} = \frac{1}{K} \sum_{\ell=-m_t}^{m_t} \boldsymbol{d}_x(\tilde{f}_{k,\ell}) \boldsymbol{d}_x^H(\tilde{f}_{k,\ell}), \tag{42}$$

$$\hat{\boldsymbol{S}}_{yk} = \frac{1}{K} \sum_{\ell=-m_t}^{m_t} \boldsymbol{d}_y(\tilde{f}_{k,\ell}) \boldsymbol{d}_y^H(\tilde{f}_{k,\ell}) \tag{43}$$

where $\hat{\boldsymbol{S}}_{xk}$ and $\hat{\boldsymbol{S}}_{yk}$ represent PSD estimators at frequency $\tilde{f}_k$ using unweighted frequency-domain smoothing [46]. By the local smoothness assumption (A2), for $\ell = -m_t, -m_t+1, \cdots, m_t$, we have

$$\boldsymbol{d}_x(\tilde{f}_{k,\ell}) \sim \mathcal{N}_c(\boldsymbol{0}, \boldsymbol{S}_x(\tilde{f}_k)), \tag{44}$$

$$\boldsymbol{d}_y(\tilde{f}_{k,\ell}) \sim \mathcal{N}_c(\boldsymbol{0}, \boldsymbol{S}_y(\tilde{f}_k)). \tag{45}$$

2) MULTIPLE COMPLEX DIFFERENTIAL GRAPHS

To lighten notation, henceforth we will denote the true values of $\boldsymbol{S}_x(\tilde{f}_k)$ and $\boldsymbol{S}_y(\tilde{f}_k)$ as $\boldsymbol{S}_{xk}^\diamond$ and $\boldsymbol{S}_{yk}^\diamond$, respectively, with their respective sample estimates $\hat{\boldsymbol{S}}_{xk}$ and $\hat{\boldsymbol{S}}_{yk}$, $k \in [M]$.

Our objective is to ascertain if $\boldsymbol{\Delta}(f) = \boldsymbol{S}_y^{-1}(f) - \boldsymbol{S}_x^{-1}(f) = \boldsymbol{0} \forall f \in [0,1]$. Since the PSD matrix $\boldsymbol{S}(f)$ of any real discrete-time zero-mean stationary random process is periodic with period one and $\boldsymbol{S}(-f) = \boldsymbol{S}^H(f)$, it is enough to check if $\boldsymbol{\Delta}(f) = \boldsymbol{0} \forall f \in [0, 0.5]$, and therefore, in the associated differential graph $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$ we have $\{i,j\} \notin \mathcal{E}_\Delta$ iff $[\boldsymbol{\Delta}(f)]_{ij} = 0$ for every $f \in [0, 0.5]$. Let $\boldsymbol{\Delta}_k = \boldsymbol{S}_{yk}^{-1} - \boldsymbol{S}_{xk}^{-1}$ and $\boldsymbol{\Delta}_k^\diamond = (\boldsymbol{S}_{yk}^\diamond)^{-1} - (\boldsymbol{S}_{xk}^\diamond)^{-1}$. Under assumption (A2) (and recalling that we exclude frequency points f_0 and $f_{n/2}$), $\boldsymbol{\Delta}(f)$, $\forall f \in [0, 0.5]$ translates to $\boldsymbol{\Delta}_k$, $k \in [M]$ such that $\{\boldsymbol{\Delta}(f) = \boldsymbol{0},\ \forall f \in [0, 0.5]\}$ is equivalent to $\{\boldsymbol{\Delta}_k = \boldsymbol{0},\ k \in [M]\}$, and therefore, in the associated differential graph $\mathcal{G}_\Delta = (V, \mathcal{E}_\Delta)$ we have $\{i,j\} \notin \mathcal{E}_\Delta$ iff $[\boldsymbol{\Delta}_k]_{ij} = 0$ for every $k \in [M]$.

Observe that for any fixed $k \in [M]$, $\boldsymbol{\Delta}_k^\diamond$ characterizes a complex differential graph (cf. Sec. II) with "precision matrix" difference $(\boldsymbol{S}_{yk}^\diamond)^{-1} - (\boldsymbol{S}_{xk}^\diamond)^{-1}$. To estimate $\boldsymbol{\Delta}_k^\diamond$ we have $K = 2m_t + 1$ independent complex measurements $\boldsymbol{d}_x(\tilde{f}_{k,\ell})$ and $\boldsymbol{d}_y(\tilde{f}_{k,\ell})$, $\ell = -m_t, -m_t+1, \cdots, m_t$. The D-trace loss for the kth differential graph is $L(\boldsymbol{\Delta}_k, \hat{\boldsymbol{S}}_{xk}, \hat{\boldsymbol{S}}_{yk})$ with $L(\cdot,\cdot,\cdot)$ specified by (10). Moreover, $L(\boldsymbol{\Delta}_k, \boldsymbol{S}_{xk}^\diamond, \boldsymbol{S}_{yk}^\diamond)$ is strictly convex in $\boldsymbol{\Delta}_k$ with a unique minimum at $\boldsymbol{\Delta}_k^\diamond$ (see Sec. II-B).

3) D-TRACE LOSS

Now we wish to ascertain if $[\boldsymbol{\Delta}_k]_{ij} = 0$ for every $k \in [M]$ which calls for joint consideration of all M complex differential graphs. Define

$$\tilde{\mathbf{\Delta}} := [\mathbf{\Delta}_1,\ \mathbf{\Delta}_2,\ \cdots,\ \mathbf{\Delta}_M] \in \mathbb{C}^{p\times pM} . \tag{46}$$

In order to exploit every $k \in [M]$, we propose the cost

$$\tilde{L}(\tilde{\mathbf{\Delta}}) = \sum_{k=1}^{M} L(\mathbf{\Delta}_k, \hat{\mathbf{S}}_{xk}, \hat{\mathbf{S}}_{yk}) . \tag{47}$$

Define

$$\tilde{\mathbf{\Delta}}^{(ij)} := \left[[\mathbf{\Delta}_1]_{ij},\ [\mathbf{\Delta}_2]_{ij},\ \cdots,\ [\mathbf{\Delta}_M]_{ij}\right]^\top \in \mathbb{C}^M . \tag{48}$$

Substitute $\hat{\mathbf{S}}_{yk} = \mathbf{S}^\diamond_{yk}$ and $\hat{\mathbf{S}}_{xk} = \mathbf{S}^\diamond_{xk}$ in $\tilde{L}(\tilde{\mathbf{\Delta}})$ and denote it by $\tilde{L}(\tilde{\mathbf{\Delta}}^\diamond)$. Then $\tilde{L}(\tilde{\mathbf{\Delta}}^\diamond)$, being a sum of strictly convex functions (cf. Sec. II-B), is strictly convex and has a unique minimum at $\mathbf{\Delta}_k = \mathbf{\Delta}^\diamond_k$, $k \in [M]$ (cf. Sec. II-B). Since data-based $\tilde{L}(\tilde{\mathbf{\Delta}})$ is a sum of convex functions $L(\mathbf{\Delta}_k, \hat{\mathbf{S}}_{xk}, \hat{\mathbf{S}}_{yk})$ (cf. Sec. II-B), it is convex, but not necessarily strictly convex.

Since true values $\mathbf{\Delta}^\diamond_k$'s are unavailable, our objective then is to estimate $\tilde{\mathbf{\Delta}}$ by minimizing data-based $\tilde{L}(\tilde{\mathbf{\Delta}})$ with resulting estimate

$$\hat{\tilde{\mathbf{\Delta}}} = [\hat{\mathbf{\Delta}}_1,\ \hat{\mathbf{\Delta}}_2,\ \cdots,\ \hat{\mathbf{\Delta}}_M] . \tag{49}$$

We estimate the edgeset $\mathcal{E}_\Delta$ of the differential time-series graph as

$$\hat{\mathcal{E}}_\Delta = \left\{ \{i,j\} \ :\ \|\hat{\tilde{\mathbf{\Delta}}}^{(ij)}\| > 0 \right\} . \tag{50}$$

IV. PENALIZED D-TRACE LOSS

In the high-dimensional case of $K < p$, to enforce sparsity and to make the problem well-conditioned, we propose to minimize a group penalized version of $\tilde{L}(\tilde{\mathbf{\Delta}})$ w.r.t. $\mathbf{\Delta}_k$s, given by

$$L_f(\tilde{\mathbf{\Delta}}) = \tilde{L}(\tilde{\mathbf{\Delta}}) + \sum_{i,j=1}^{p} h_\lambda\left(\|\tilde{\mathbf{\Delta}}^{(ij)}\|\right) \tag{51}$$

where, for $u \in \mathbb{R}$, $h_\lambda(u)$ is a penalty function that is a function of $|u|$. The penalty operates on the group $\tilde{\mathbf{\Delta}}^{(ij)} \in \mathbb{C}^M$, instead of individual elements of $\tilde{\mathbf{\Delta}}$.

The following penalty functions are considered:

(1) *Lasso*. For some $\lambda > 0$, $h_\lambda(u) = \lambda|u|$, $u \in \mathbb{R}$. It is a convex function that is widely used.

(2) *Log-sum*. For some $\lambda > 0$ and $1 \gg \epsilon > 0$, $h_\lambda(u) = \lambda\epsilon \ln\left(1 + \frac{|u|}{\epsilon}\right)$. It is a nonconvex function.

(3) *SCAD*. For some $\lambda > 0$ and $a > 2$, $h_\lambda(u) = \lambda|u|$ for $|u| \le \lambda$, $= (2a\lambda|u| - |u|^2 - \lambda^2)/(2(a-1))$ for $\lambda < |u| < a\lambda$, and $= \lambda^2(a+1)/2$ for $|u| \ge a$. It is a nonconvex function.

In [47] the log-sum penalty is defined as $h_\lambda(u) = \ln(1+\lambda|u|)$ whereas in [17], it is defined as $h_\lambda(u) = \lambda \ln\left(1 + \frac{|u|}{\epsilon}\right)$. We follow [17] but modify it so that, as for the lasso and SCAD penalties, our $h_\lambda(u)$ yields $\lim_{u\to 0^+} h'_\lambda(u) = \lambda$ where $h'_\lambda(u) := \frac{dh_\lambda(u)}{du}$.

V. OPTIMIZATION

The objective function $L_f(\tilde{\mathbf{\Delta}})$ is non-convex in $\mathbf{\Delta}$ for the non-convex SCAD and log-sum penalties, and convex for the lasso penalty. Now we discuss an ADMM approach, following the ADMM approach discussed in Sec. II-C for lasso, to attain a local minimum of $L_f(\tilde{\mathbf{\Delta}})$ for the non-convex SCAD and log-sum penalties, and a global minimum for the lasso penalty.

For non-convex $h_\lambda(u)$, we use a local linear approximation (LLA) (as in [19], [22]), to yield

$$h_\lambda(u) \approx h_\lambda(|u_0|) + h'_\lambda(|u_0|)(|u| - |u_0|) \ \to\ h'_\lambda(|u_0|)|u| , \tag{52}$$

where $h'(x) = dh(x)/dx$, u_0 is an initial guess, and the gradient of the penalty function is

$$h'_\lambda(|u_0|) = \begin{cases} \frac{\lambda\epsilon}{|u_0|+\epsilon} \text{ for log-sum,} \\ \begin{cases} \lambda, & \text{if } |u| \le \lambda \\ \frac{a\lambda - |u|}{a-1}, & \text{if } \lambda < |u| \le a\lambda \\ 0, & \text{if } a\lambda < |u| \end{cases} \\ \quad\text{for SCAD.} \end{cases} \tag{53}$$

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

$$h_\lambda(u) \ \to\ h'_\lambda(|u_0|)\, |u| . \tag{54}$$

By [22, Theorem 1], the LLA provides a majorization of the non-convex penalty, thereby yielding a majorization-minimization approach. By [22, Theorem 2], the LLA is the best convex majorization of the LSP and SCAD penalties.

With some initial guess $\bar{\tilde{\mathbf{\Delta}}} = [\bar{\mathbf{\Delta}}_1,\ \bar{\mathbf{\Delta}}_2,\ \cdots,\ \bar{\mathbf{\Delta}}_M]$, in LSP we replace

$$h_\lambda\left(\|\tilde{\mathbf{\Delta}}^{(ij)}\|\right) \to \lambda_{ij} := \frac{\lambda\epsilon}{\|\bar{\tilde{\mathbf{\Delta}}}^{(ij)}\| + \epsilon} . \tag{55}$$

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

$$\lambda_{ij} = \begin{cases} \lambda, & \text{if } \|\bar{\tilde{\mathbf{\Delta}}}^{(ij)}\| \le \lambda \\ \frac{a\lambda - \|\bar{\tilde{\mathbf{\Delta}}}^{(ij)}\|}{a-1}, & \text{if } \lambda < \|\bar{\tilde{\mathbf{\Delta}}}^{(ij)}\| \le a\lambda \\ 0, & \text{if } a\lambda < \|\bar{\tilde{\mathbf{\Delta}}}^{(ij)}\| \end{cases} . \tag{56}$$

With LLA, the original objective function is transformed to its convex LLA approximation

$$\tilde{L}_f(\tilde{\mathbf{\Delta}}) = \tilde{L}(\tilde{\mathbf{\Delta}}) + \sum_{i,j=1}^{p} \lambda_{ij} \|\tilde{\mathbf{\Delta}}^{(ij)}\| . \tag{57}$$

For lasso, we have $\lambda_{ij} = \lambda \forall i,j$. If $\bar{\tilde{\mathbf{\Delta}}}^{(ij)} = \mathbf{0}$, we obtain $\lambda_{ij} = \lambda \forall i,j$.

We follow an ADMM approach following the ADMM approach discussed in Sec. II-C for lasso, for both lasso and LLA to LSP/SCAD. For non-convex penalties, we have an iterative solution: first solve with lasso penalty, then use this solution for initialization to LLA and solve again the LLA convex problem. In practice, just two iterations seem to be enough. Using variable splitting and adding the equality constraint $\tilde{\boldsymbol{W}} = \tilde{\boldsymbol{\Delta}}$ with $\tilde{\boldsymbol{W}} = [\boldsymbol{W}_1 \ \cdots \ \boldsymbol{W}_M]$, $\boldsymbol{W}_k \in \mathbb{C}^{p\times p}$ for $k \in [M]$, consider ($\tilde{\boldsymbol{W}}^{(ij)}$ is defined similar to (48))

$$\min_{\tilde{\boldsymbol{\Delta}}, \tilde{\boldsymbol{W}}} \left\{ \tilde{L}_f(\tilde{\boldsymbol{\Delta}}) + \sum_{i,j=1}^{p} \lambda_{ij} \| \tilde{\boldsymbol{W}}^{(ij)} \| \right\}, \tag{58}$$

$$\text{subject to } \ \tilde{\boldsymbol{\Delta}} = \tilde{\boldsymbol{W}}, \tag{59}$$

where, in the penalty, we use $\tilde{\boldsymbol{W}}$ instead of $\tilde{\boldsymbol{\Delta}}$. Let $\tilde{\boldsymbol{U}} = [\boldsymbol{U}_1 \ \cdots \ \boldsymbol{U}_M]$, $\boldsymbol{U}_k \in \mathbb{C}^{p\times p}$ for $k \in [M]$, denote the dual variables and $\rho > 0$ denote the penalty parameter in the ADMM algorithm. The scaled augmented Lagrangian for this problem is [31]

$$\tilde{L}_\rho(\tilde{\boldsymbol{\Delta}}, \tilde{\boldsymbol{W}}, \tilde{\boldsymbol{U}}) = \tilde{L}_f(\tilde{\boldsymbol{\Delta}}) + \sum_{i,j=1}^{p} \lambda_{ij} \| \tilde{\boldsymbol{W}}^{(ij)} \| + \frac{\rho}{2} \sum_{k=1}^{M} \| \boldsymbol{\Delta}_k - \boldsymbol{W}_k + \boldsymbol{U}_k \|_F^2 . \tag{60}$$

Following Sec. II-C, given the results $\tilde{\boldsymbol{\Delta}}^{(m)}, \tilde{\boldsymbol{W}}^{(m)}, \tilde{\boldsymbol{U}}^{(m)}$ of the mth iteration, in the $(m+1)$st iteration, an ADMM algorithm executes the following three updates:

(i) $\tilde{\boldsymbol{\Delta}}^{(m+1)} \leftarrow \arg\min_{\tilde{\boldsymbol{\Delta}}} \sum_{k=1}^{M} \tilde{L}_{ak}(\boldsymbol{\Delta}_k)$ where

$$\tilde{L}_{ak}(\boldsymbol{\Delta}_k) = L(\boldsymbol{\Delta}_k, \hat{\boldsymbol{S}}_{xk}, \hat{\boldsymbol{S}}_{yk}) + \frac{\rho}{2} \| \boldsymbol{\Delta}_k - \boldsymbol{W}_k^{(m)} + \boldsymbol{U}_k^{(m)} \|_F^2 .$$

(ii) $\tilde{\boldsymbol{W}}^{(m+1)} \leftarrow \arg\min_{\tilde{\boldsymbol{W}}} \tilde{L}_b(\tilde{\boldsymbol{W}})$ where

$$\tilde{L}_b(\tilde{\boldsymbol{W}}) = \frac{\rho}{2} \sum_{k=1}^{M} \| \boldsymbol{\Delta}_k^{(m+1)} - \boldsymbol{W}_k + \boldsymbol{U}_k^{(m)} \|_F^2 + \sum_{i,j=1}^{p} \lambda_{ij} \| \tilde{\boldsymbol{W}}^{(ij)} \| .$$

(iii) $\tilde{\boldsymbol{U}}^{(m+1)} \leftarrow \tilde{\boldsymbol{U}}^{(m)} + \tilde{\boldsymbol{\Delta}}^{(m+1)} - \tilde{\boldsymbol{W}}^{(m+1)}$.

Some details regarding updates (i) and (ii) are given below.

Update (i): The optimization in step (i) is separable in $\boldsymbol{\Delta}_k$, and the solution discussed in Sec. II-C applies. Perform the eigen-decomposition of $\hat{\boldsymbol{S}}_{xk}$ and $\hat{\boldsymbol{S}}_{yk}$ as $\hat{\boldsymbol{S}}_{xk} = \boldsymbol{Q}_{xk}\boldsymbol{D}_{xk}\boldsymbol{Q}_{xk}^H$ and $\hat{\boldsymbol{S}}_{yk} = \boldsymbol{Q}_{yk}\boldsymbol{D}_{yk}\boldsymbol{Q}_{yk}^H$ where $\boldsymbol{D}_{xk}$ and $\boldsymbol{D}_{yk}$ are diagonal matrices, $\boldsymbol{Q}_{xk}\boldsymbol{Q}_{xk}^H = \boldsymbol{I}_p$ and $\boldsymbol{Q}_{yk}\boldsymbol{Q}_{yk}^H = \boldsymbol{I}_p$. Define a matrix $\boldsymbol{D}^{(k)} \in \mathbb{R}^{p\times p}$ that organizes the diagonal of $(\boldsymbol{D}_{yk} \otimes \boldsymbol{D}_{xk} + \frac{\rho}{2}\boldsymbol{I}_{p^2})^{-1}$ in a matrix with the (i,j)th element $[\boldsymbol{D}^{(k)}]_{ij} = 1/([\boldsymbol{D}_{xk}]_{ii}[\boldsymbol{D}_{yk}]_{jj} + \frac{\rho}{2})$. Then, for $k \in [M]$, we have

$$\boldsymbol{\Delta}_k^{(m+1)} = \boldsymbol{Q}_{xk}\Big[\boldsymbol{D}^{(k)} \circ \big[\boldsymbol{Q}_{xk}^H(\hat{\boldsymbol{S}}_{xk} - \hat{\boldsymbol{S}}_{yk} + \frac{\rho}{2}(\boldsymbol{W}_k^{(m)} - \boldsymbol{U}_k^{(m)}))\boldsymbol{Q}_{yk}\big]\Big]\boldsymbol{Q}_{yk}^H . \tag{61}$$

Update (ii): The optimization in step (ii) is separable in $\tilde{\boldsymbol{W}}^{(ij)}$, and the solution discussed in Sec. II-C applies with $\lambda \to \lambda_{ij}$. With $\boldsymbol{A}_k = \boldsymbol{\Delta}_k^{(m+1)} + \boldsymbol{U}_k^{(m)}$, $(b)_+ = \max(0, b)$, and for $k \in [M]$ and $i,j \in [p]$,

$$[\boldsymbol{W}_k^{(m+1)}]_{ij} = \left(1 - \frac{\lambda_{ij}}{\rho \|\tilde{\boldsymbol{A}}^{(ij)}\|}\right)_+ [\boldsymbol{A}_k]_{ij} , \tag{62}$$

$$\text{where } \tilde{\boldsymbol{A}}^{(ij)} = \big[[\boldsymbol{A}_1]_{ij}, \ \cdots, \ [\boldsymbol{A}_M]_{ij}\big]^\top \in \mathbb{C}^M . \tag{63}$$

Algorithm 1 ADMM Algorithm for Solving (60)

```
Input: PSD estimators $\hat{S}_{xk}$ and $\hat{S}_{yk}$, $k \in [M]$ (computed using (42) and (43)), regularization and penalty parameters $\lambda_{ij}$ $(i,j \in [p])$ and $\rho = \bar{\rho}$, tolerances $\tau_{abs}$ and $\tau_{rel}$, variable penalty factor $\bar{\mu}$, maximum number of iterations $m_{\max}$. Initial guess $\bar{\Delta}_k$, $k \in [M]$.
Output: Estimated $\hat{\Delta}_k$, $k \in [M]$.
 1: Initialize: $\Delta_k^{(0)} = \bar{\Delta}_k, U_k^{(0)} = W_k^{(0)} = 0, k \in [M], \rho^{(0)} = \bar{\rho}$
 2: Eigen-decompose $\hat{S}_{xk}$ and $\hat{S}_{yk}$ as $\hat{S}_{xk} = Q_{xk}D_{xk}Q_{xk}^H$ and $\hat{S}_{yk} = Q_{yk}D_{yk}Q_{yk}^H$, $k \in [M]$.
 3: converged = false, $m = 0$
 4: while converged = false and $m \le m_{\max}$, do
 5:     For $k \in [M]$, construct $D^{(k)} \in \mathbb{R}^{p\times p}$ with $[D^{(k)}]_{ij} = 1/([D_{xk}]_{ii}[D_{yk}]_{jj} + \frac{\rho^{(m)}}{2})$.
 6:     For $k \in [M]$, set $\Delta_k^{(m+1)} = Q_{xk}\Big[D^{(k)} \circ \big[Q_{xk}^H(\hat{S}_{xk} - \hat{S}_{yk} + \frac{\rho}{2}(W_k^{(m)} - U_k^{(m)}))Q_{yk}\big]\Big]Q_{yk}^H$.
 7:     For $k \in [M]$, define $A_k = \Delta_k^{(m+1)} + U_k^{(m)}$ and $\tilde{A}^{(ij)} = \big[[A_1]_{ij}, \cdots, [A_M]_{ij}\big]^\top$. For $k \in [M]$ and $i,j \in [p]$, $[W_k^{(m+1)}]_{ij} = \left(1 - \frac{\lambda_{ij}}{\rho^{(m)} \|\tilde{A}^{(ij)}\|}\right)_+ [A_k]_{ij}$.
 8:     Dual update $U_k^{(m+1)} \leftarrow U_k^{(m)} + \Delta_k^{(m+1)} - W_k^{(m+1)}$, $k \in [M]$.
 9:     Check convergence. With $e_1$, $e_2$, $e_3$, $R_p^{(m+1)}$, $R_d^{(m+1)}$, $\tau_{pri}$ and $\tau_{dual}$ as defined in (64)-(70), respectively, let $d_p = \|R_p^{(m+1)}\|_F$ and $d_d = \|R_d^{(m+1)}\|_F$. If $(d_p \le \tau_{pri})$ and $(d_d \le \tau_{dual})$, set converged = true.
10:     Update penalty parameter $\rho$ :
        $\rho^{(m+1)} = \begin{cases} 2\rho^{(m)} & \text{if } d_p > \bar{\mu}\, d_d \\ \rho^{(m)}/2 & \text{if } d_d > \bar{\mu}\, d_p \\ \rho^{(m)} & \text{otherwise}. \end{cases}$
        We also need to set $U^{(m+1)} = U^{(m+1)}/2$ for $d_p > \bar{\mu} d_d$ and $U^{(m+1)} = 2U^{(m+1)}$ for $d_d > \bar{\mu} d_p$.
11:     $m \leftarrow m + 1$
12: end while
13: With $\Delta_k$, $k \in [M]$, denoting the converged estimates, set $\hat{\Delta}_k = (\Delta_k + \Delta_k^H)/2$, $k \in [M]$, and
        $\hat{\tilde{\Delta}} = [\hat{\Delta}_1, \ \hat{\Delta}_2, \ \cdots, \ \hat{\Delta}_M]$.
```

A pseudocode for the ADMM algorithm to solve (60) is given in Algorithm 1 where we use the stopping (convergence) criterion following [31, Sec. 3.3.1] and varying penalty parameter ρ following [31, Sec. 3.4.1]. The variables defined in (64)-(70) are needed in Algorithm 1 with $\boldsymbol{\Delta}_k^{(m+1)}$, $\boldsymbol{W}_k^{(m+1)}$, $\boldsymbol{U}_k^{(m+1)}$ as defined therein:

$$e_1 = \| [\boldsymbol{\Delta}_1^{(m+1)}, \cdots, \boldsymbol{\Delta}_M^{(m+1)}] \|_F \tag{64}$$

$$e_2 = \| [\boldsymbol{W}_1^{(m+1)}, \cdots, \boldsymbol{W}_M^{(m+1)}] \|_F \tag{65}$$

$$e_3 = \|[\boldsymbol{U}_1^{(m+1)}, \cdots, \boldsymbol{U}_M^{(m+1)}]\|_F \tag{66}$$

$$\boldsymbol{R}_p^{(m+1)} = \left[\boldsymbol{\Delta}_1^{(m+1)} - \boldsymbol{W}_1^{(m+1)}, \ \cdots, \ \boldsymbol{\Delta}_M^{(m+1)} - \boldsymbol{W}_M^{(m+1)}\right] \tag{67}$$

$$\boldsymbol{R}_d^{(m+1)} = \rho^{(m)}\left[\boldsymbol{W}_1^{(m+1)} - \boldsymbol{W}_1^{(m)}, \ \cdots, \ \boldsymbol{W}_M^{(m+1)} - \boldsymbol{W}_M^{(m)}\right] \tag{68}$$

$$\tau_{pri} = p\sqrt{M}\,\tau_{abs} + \tau_{rel}\ \max(e_1, e_2) \tag{69}$$

$$\tau_{dual} = p\sqrt{M}\,\tau_{abs} + \tau_{rel}\, e_3/\rho^{(m)}\,. \tag{70}$$

Our overall ADMM-based optimization algorithm is as follows.

1. Given M and $K = 2m_t + 1$, calculate $\hat{\boldsymbol{S}}_{xk}$ and $\hat{\boldsymbol{S}}_{yk}$, $k \in [M]$ (computed using (42) and (43)). Initialize iteration $\tilde{m} = 0$, $\tilde{\boldsymbol{\Delta}}^{(0)} = \boldsymbol{0}$, $\bar{\bar{\boldsymbol{\Delta}}} = [\bar{\boldsymbol{\Delta}}_1, \ \bar{\boldsymbol{\Delta}}_2, \ \cdots, \ \bar{\boldsymbol{\Delta}}_M] = \tilde{\boldsymbol{\Delta}}^{(0)}$ and use $\bar{\bar{\boldsymbol{\Delta}}}$ to compute λ_{ij}'s.
2. Execute Algorithm 1 with initial guess $\bar{\bar{\boldsymbol{\Delta}}}$. Denote the resulting estimate by $\hat{\hat{\boldsymbol{\Delta}}}$. Let $\tilde{m} \leftarrow \tilde{m} + 1$.
3. Quit if using lasso, else set $\tilde{\boldsymbol{\Delta}}^{(\tilde{m})} = \hat{\hat{\boldsymbol{\Delta}}}$ and $\bar{\bar{\boldsymbol{\Delta}}} = \tilde{\boldsymbol{\Delta}}^{(\tilde{m})}$ to re-compute λ_{ij}'s via the LLA.
4. Repeat steps 2 and 3 until convergence.

A pseudocode for the above ADMM algorithm is given in Algorithm 2. It utilizes Algorithm 1 in each LLA step.

For the numerical results in Secs. VII and VIII, we used $\bar{\mu} = 10$, $\bar{\rho} = 2$, $\epsilon = 0.001$ for log-sum penalty, $a = 3.7$ (as in [18], [19]) for the SCAD penalty, $\tau_{abs} = \tau_{rel} = 10^{-4}$, $m_{\max} = 200$, and $\tilde{m}_{\max} = 1$ for lasso and $= 2$ for LSP and SCAD penalties.

Algorithm 2 LLA-Based ADMM Algorithm for Optimizing (51)

Input: PSD estimators $\hat{\boldsymbol{S}}_{xk}$ and $\hat{\boldsymbol{S}}_{yk}$, $k \in [M]$ (computed using (42) and (43)), regularization and penalty parameters λ and $\rho = \bar{\rho}$, tolerances τ_{abs} and τ_{rel}, variable penalty factor $\bar{\mu}$, maximum number of iterations $\tilde{m}_{\max}$. For lasso penalty, $\tilde{m}_{\max} = 1$

Output: Estimated $\hat{\boldsymbol{\Delta}}_k$, $k \in [M]$, and edge-set $\hat{\mathcal{E}}_\Delta$

```
1:  Initialize $\tilde{\boldsymbol{\Delta}}^{(0)} = \boldsymbol{0}$ and $\bar{\bar{\boldsymbol{\Delta}}} = [\bar{\boldsymbol{\Delta}}_1, \bar{\boldsymbol{\Delta}}_2, \cdots, \bar{\boldsymbol{\Delta}}_M] = \tilde{\boldsymbol{\Delta}}^{(0)} \in \mathbb{C}^{p\times pM}$. Set $\lambda_{ij} = \lambda$, $i,j \in [p]$.
2:  $\tilde{m} = 1$
3:  while $\tilde{m} \le \tilde{m}_{\max}$, do
4:      Execute Algorithm 1, resulting in output $\hat{\hat{\boldsymbol{\Delta}}}$. Set $\tilde{\boldsymbol{\Delta}}^{(\tilde{m})} = \hat{\hat{\boldsymbol{\Delta}}}$.
5:      if LSP/SCAD then
6:          Set $\bar{\bar{\boldsymbol{\Delta}}} = \tilde{\boldsymbol{\Delta}}^{(\tilde{m})}$ and re-compute $\lambda_{ij}$'s via the LLA (55) or (56).
7:      end if
8:      $\tilde{m} \leftarrow \tilde{m} + 1$
9:  end while
10: With $\boldsymbol{\Delta}_k$, $k \in [M]$, denoting the converged estimates, set $\hat{\boldsymbol{\Delta}}_k = (\boldsymbol{\Delta}_k + \boldsymbol{\Delta}_k^H)/2$, $k \in [M]$, and
        $\hat{\hat{\boldsymbol{\Delta}}} = [\hat{\boldsymbol{\Delta}}_1, \ \hat{\boldsymbol{\Delta}}_2, \ \cdots, \ \hat{\boldsymbol{\Delta}}_M]$.
    If $\|\hat{\hat{\boldsymbol{\Delta}}}^{(ij)}\| > 0$ assign edge $\{i,j\} \in \hat{\mathcal{E}}_\Delta$, else $\{i,j\} \notin \hat{\mathcal{E}}_\Delta$.
```

A. CONVERGENCE

The LLA-based objective function $\tilde{L}_f(\tilde{\boldsymbol{\Delta}})$, given by (57), is convex in $\tilde{\boldsymbol{\Delta}}$ (cf. Sec. III-A3). It is also closed, proper and lower semi-continuous. Hence, for any fixed $\rho > 0$, the ADMM algorithm is guaranteed to converge [31, Sec. 3.2 and Appendix A], in the sense that we have primal residual (67) convergence to 0, dual residual (68) convergence to 0, and the objective function $\tilde{L}_f(\tilde{\boldsymbol{\Delta}})$ convergence to the optimal value.

B. BIC FOR TUNING PARAMETER SELECTION

Given n and the chosen K and M, for model selection we follow a BIC-like criterion similar to as given in [8, Sec. III-E] (which follows [5] who invokes [12]) for time-domain approaches. Let $|\boldsymbol{A}|_0$ denote the number of nonzero elements in $\boldsymbol{A}$ and suppose that $\hat{\hat{\boldsymbol{\Delta}}} = [\hat{\boldsymbol{\Delta}}_1, \ \hat{\boldsymbol{\Delta}}_2, \ \cdots, \ \hat{\boldsymbol{\Delta}}_M]$ minimizes (51). We choose λ to minimize BIC(λ) given by

$$\mathrm{BIC}(\lambda) = 4K \sum_{k=1}^{M} \|\hat{\boldsymbol{S}}_{xk}\,\hat{\boldsymbol{\Delta}}_k \hat{\boldsymbol{S}}_{yk} - (\hat{\boldsymbol{S}}_{xk} - \hat{\boldsymbol{S}}_{yk})\|_F + \ln(4K) \sum_{k=1}^{M} |\hat{\boldsymbol{\Delta}}_k|_0\,. \tag{71}$$

Following [5] we use the term BIC (Bayesian information criterion) for it even though the cost function used is not negative log-likelihood although $\ln(4K)\sum_{k=1}^{M}\|\hat{\boldsymbol{\Delta}}_k\|_0$ penalizes over-parametrization as in BIC. It is based on the fact that true $\boldsymbol{\Delta}_k^\diamond$ satisfies $\boldsymbol{S}_{xk}^\diamond \boldsymbol{\Delta}_k^\diamond \boldsymbol{S}_{yk}^\diamond - (\boldsymbol{S}_{xk}^\diamond - \boldsymbol{S}_{yk}^\diamond) = \boldsymbol{0}$. Since (71) is not scale invariant, we scale both $\hat{\boldsymbol{S}}_{xk}$ and $\hat{\boldsymbol{S}}_{yk}$ (and $\hat{\boldsymbol{\Delta}}_k$ commensurately) by $\bar{\boldsymbol{\Sigma}}^{-1}$ where $\bar{\boldsymbol{\Sigma}} = \mathrm{diag}\{\hat{\boldsymbol{\Sigma}}_x\}$ is a diagonal matrix of diagonal elements of $\hat{\boldsymbol{\Sigma}}_x = \frac{1}{n_x}\sum_{t=1}^{n_x} \boldsymbol{x}(t)\boldsymbol{x}^\top(t)$ (we have $n_x = n_y = n$ in this paper). We have M models, each with K complex measurements $\boldsymbol{d}_x(\tilde{f}_{k,\ell})$ and $\boldsymbol{d}_y(\tilde{f}_{k,\ell})$, leading to $4K$ real samples for each model: (71) reflects that.

In our numerical results we search over $\lambda \in [\lambda_\ell, \lambda_u]$, where λ_ℓ and λ_u are selected via a heuristic as in [8]. Find the smallest λ, labeled λ_{sm} for which we get a no-edge model; then we set $\lambda_u = \lambda_{sm}/2$ and $\lambda_\ell = \lambda_u/10$.

VI. THEORETICAL ANALYSIS

Here we analyze some properties (consistency in inverse PSD difference estimation and graph recovery) of the minimizer of the convex function $\tilde{L}_f(\tilde{\boldsymbol{\Delta}})$ specified by (57), by following the approach of [20]. The approach of [20] requires $\lambda_{ij} > 0$ for every $i,j \in p$, a condition that is violated by the SCAD penalty. Therefore, our theoretical analysis applies to the lasso and the log-sum penalties only.

Define the true differential edgeset $\mathcal{E}_\Delta^\diamond$ and its cardinality s,

$$\mathcal{E}_\Delta^\diamond = \Big\{\{i,j\} \ : \ [(\boldsymbol{S}_y^\diamond(f))^{-1} - (\boldsymbol{S}_y^\diamond(f))^{-1}]_{ij} \neq 0, \quad 0 \le f \le 0.5\Big\}, \quad s = |\mathcal{E}_\Delta^\diamond|\,. \tag{72}$$

In the rest of this section, we allow p, $K = 2m_t + 1$, M, s and λ to be a functions of sample size n, denoted as p_n, K_n, M_n, s_n and λ_n, respectively. With $\tau > 2$, define

$$B_{xy} = \max_{f \in [0,0.5]} \left\{ \|S_x^\diamond(f)\|_\infty , \|S_y^\diamond(f)\|_\infty \right\}, \tag{73}$$

$$B_d = \max_{f \in [0,0.5]} \|(S_y^\diamond(f))^{-1} - (S_x^\diamond(f))^{-1}\|_\infty , \tag{74}$$

$$\phi_{\min}^\diamond = \min_{f \in [0,0.5]} \{\phi_{\min}(S_x^\diamond(f)) \times \phi_{\min}(S_y^\diamond(f))\}, \tag{75}$$

$$\sigma_{xy} = \max_{f \in [0,0.5], \ell \in [p]} \left\{ [S_x^\diamond(f)]_{\ell\ell} , [S_y^\diamond(f)]_{\ell\ell} \right\}, \tag{76}$$

$$C_0 = 80\, \sigma_{xy} \sqrt{2\big(\ln(16 p_n^\tau M_n)/\ln(p_n)\big)} , \tag{77}$$

$$N_1 = \arg\min_n \left\{ n \;:\; K_n > 2 \ln(16 p_n^\tau M_n) \right\}, \tag{78}$$

$$N_2 = \arg\min_n \left\{ n \;:\; K_n > C_0^2 \ln(p_n)/B_{xy} \right\}, \tag{79}$$

$$N_3 = \arg\min_n \Big\{ n \;:\; \sqrt{K_n}/M_n \geq 768\, B_{xy} B_{\text{init}}^2 s_n C_0 \sqrt{\ln(p_n)}/\phi_{\min}^\diamond \Big\}, \tag{80}$$

$$B_{\text{init}} = \begin{cases} 1 & : \text{lasso} \\ 1 + \max_{i,j \in [p]} \|\bar{\tilde{\Delta}}^{(ij)}\|/\epsilon & : \text{log-sum} \end{cases} \tag{81}$$

where $\bar{\tilde{\Delta}}$ is the initialization for LLA to the log-sum penalty (see (55)).

Let $\hat{\tilde{\Delta}} = \arg\min_{\tilde{\Delta}} \tilde{L}_f(\tilde{\Delta})$ where $\tilde{L}_f(\tilde{\Delta})$ is specified in (57). The proof of Theorem 1 is given in the Appendix B.

Theorem 1: Under assumptions (A1)-(A2), if

$$\lambda_n \geq 2B_{\text{init}} \sqrt{M_n}\big(6\, B_{xy} B_d s_n + 4\big) C_0 \sqrt{\frac{\ln(p_n)}{K_n}}, \tag{82}$$

$$n \geq \max\{N_1, N_2, N_3\}, \tag{83}$$

then with probability $> 1 - 2/p_n^{\tau-2}$, we have

$$\|\hat{\tilde{\Delta}} - \tilde{\Delta}^\diamond\|_F = \sqrt{\sum_{k=1}^{M_n} \|\hat{\Delta}_k - \Delta_k^\diamond\|_F^2} \leq \frac{4\sqrt{s_n}\, \lambda_n}{\phi_{\min}^\diamond} \tag{84}$$

for any $\tau > 2$.

Remark 1 (Convergence Rate): If B_{xy}, σ_{xy}, $\phi_{\min}^\diamond$ and B_d stay bounded with increasing sample size n, we have $\|\hat{\tilde{\Delta}} - \tilde{\Delta}^\diamond\|_F = \mathcal{O}_P(s_n^{1.5}\sqrt{M_n \ln(p_n)/K_n})$. Therefore, for $\|\hat{\tilde{\Delta}} - \tilde{\Delta}^\diamond\|_F \to 0$ as $n \to \infty$, we must have $s_n^{1.5}\sqrt{M_n \ln(p_n)/K_n}) \to 0$. Note that $K_n M_n \approx n/2$, therefore, for $\|\hat{\tilde{\Delta}} - \tilde{\Delta}^\diamond\|_F \to 0$ as $n \to \infty$, we need $s_n^{1.5}\sqrt{n \ln(p_n)/K_n^2}) \to 0$. $\square$

We now address graph recovery. We follow the proof technique of [9, Theorem 10] in establishing Theorem 2 whose proof is in the Appendix B. For some $\gamma_n > 0$, define

$$\hat{\mathcal{E}}_\Delta = \left\{ \{i,j\} \;:\; \|\hat{\tilde{\Delta}}^{(ij)}\| > \gamma_n > 0 \right\}, \tag{85}$$

$$\tilde{\mathcal{E}}_\Delta^\diamond = \left\{ \{i,j\} \;:\; \|(\tilde{\Delta}^\diamond)^{(ij)}\| > 0 \right\}, \tag{86}$$

$$\bar{\sigma}_n = \frac{4\sqrt{s_n}\, \lambda_n}{\phi_{\min}^\diamond}, \tag{87}$$

$$\nu = \min_{\{i,j\} \in \mathcal{E}_\Delta^\diamond} \|\big((S_y^\diamond(f))^{-1} - (S_x^\diamond(f))^{-1}\big)^{(ij)}\|, \tag{88}$$

$$N_4 = \arg\min \left\{ n \;:\; \bar{\sigma}_n \leq 0.4\nu \right\}. \tag{89}$$

Theorem 2: For $\gamma_n = 0.5\nu$ and $n \geq N_4$, $\hat{\mathcal{E}}_\Delta = \tilde{\mathcal{E}}_\Delta^\diamond$ with probability $>\ 1 - 2/p_n^{\tau-2}$ under the conditions of Theorem 1.

VII. SYNTHETIC DATA EXAMPLES

We now present numerical results using synthetic data to illustrate the proposed approach (real data results are in Sec. VIII). In synthetic data examples the ground truth is known and this allows for assessment of the efficacy of various approaches in graph learning.

A. GRAPHS WITH 120 NODES

We consider two models for time-dependent data generation with $p = 120$.

1) AR MODEL

The time series data $\{\boldsymbol{x}(t)\}$, $\boldsymbol{x}(t) \in \mathbb{R}^p$, is generated using a vector autoregressive (AR) model of order 3 (VAR(3)) as follows. Let $\{\boldsymbol{w}(t)\}$, $\boldsymbol{w}(t) \in \mathbb{R}^p$, denote an i.i.d. zero-mean Gaussian sequence with precision matrix $\boldsymbol{\Omega}$ and let square matrices $\boldsymbol{A}_i \in \mathbb{R}^{p \times p}$, $i \in [3]$, be block-diagonal with 15×15 sub-blocks $\boldsymbol{A}_i^{(q)}$, $q \in [8]$. Then $\{\boldsymbol{x}(t)\}$ is generated as

$$\boldsymbol{x}(t) = \sum_{i=1}^{3} \boldsymbol{A}_i \boldsymbol{x}(t-i) + \boldsymbol{w}(t). \tag{90}$$

The diagonal entries of $\boldsymbol{\Omega}$ are set to 0.5, and the off-diagonal entries follow an Erdös-Rènyi (ER) graph with connection probability $p_{er} = 0.001$: if nodes j and k are not connected in the ER graph, we have $[\boldsymbol{\Omega}]_{jk} = 0$, and if they are connected, then $[\boldsymbol{\Omega}]_{jk}$ is uniformly distributed over $[-0.4, -0.1] \cup [0.1, 0.4]$. Only 20% of entries of $\boldsymbol{A}_i^{(q)}$'s are nonzero (randomly picked) and the nonzero elements are independently and uniformly distributed over $[-0.8, -0.3] \cup [0.3, 0.8]$. We then check if the VAR(3) model is stable with all eigenvalues of the companion matrix ≤ 0.95 in magnitude; if not, the we scale $\boldsymbol{A}_i$'s to fulfill this condition (see [27, Sec. VI.A], [48, Sec. 6.1]). To generate $\boldsymbol{y}$-data, we randomly eliminate one of the 8 clusters ($\boldsymbol{A}_i^{(q)}$'s for randomly picked q) of $\boldsymbol{x}(t)$ and replace it with an independently generated $\boldsymbol{A}_i^{(q)}$, $i \in [3]$.

2) MA MODEL

Here the time series data $\{\boldsymbol{x}(t)\}$, $\boldsymbol{x}(t) \in \mathbb{R}^p$, is generated using a vector moving average (MA) model of order 3 (MA(3)) as follows. Let $\{\boldsymbol{w}(t)\}$, $\boldsymbol{w}(t) \in \mathbb{R}^p$, with precision matrix $\boldsymbol{\Omega}$, be as for the AR model, and let square matrices $\boldsymbol{B}_i \in \mathbb{R}^{p \times p}$, $i \in [3]$, be block-diagonal with 15×15 sub-blocks $\boldsymbol{B}_i^{(q)}$, $q \in [8]$. Then $\{\boldsymbol{x}(t)\}$ is generated as

$$\boldsymbol{x}(t) = 0.5 \boldsymbol{I}_p \boldsymbol{w}(t) + \sum_{i=1}^{3} (\boldsymbol{B}_i/i) \boldsymbol{w}(t-i). \tag{91}$$

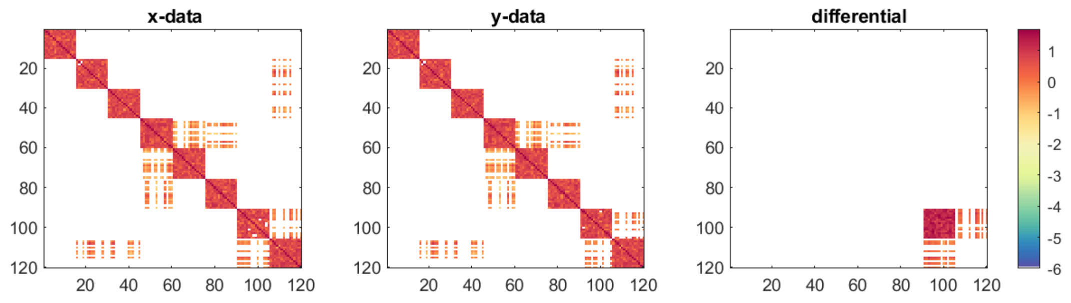


FIGURE 1. True $\log_{10}\left(\sum_{f=0:0.01:5}|[S_x^\diamond(f)]_{ij}|\right)$ (left), $\log_{10}\left(\sum_{f=0:0.01:5}|[S_y^\diamond(f)]_{ij}|\right)$ (middle), and $\log_{10}\left(\sum_{f=0:0.01:5}|[(S_y^\diamond(f))^{-1}-(S_x^\diamond(f))^{-1}]_{ij}|\right)$ (right), $i,j\in[120]$, for the AR model, for a single Monte Carlo run: $p=120$ nodes.

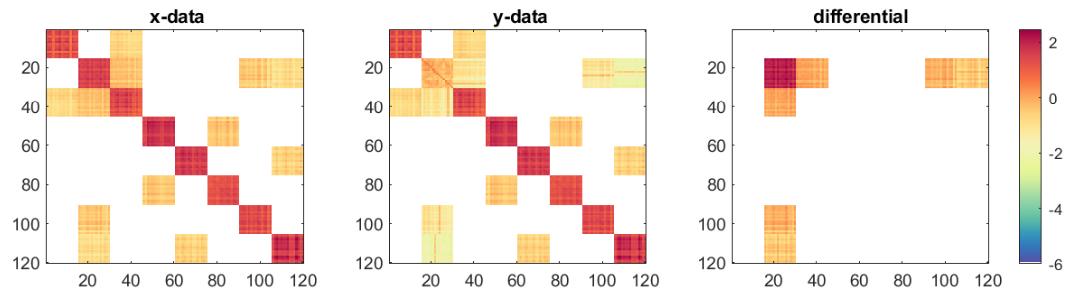


FIGURE 2. True $\log_{10}\left(\sum_{f=0:0.01:5}|[S_x^\diamond(f)]_{ij}|\right)$ (left), $\log_{10}\left(\sum_{f=0:0.01:5}|[S_y^\diamond(f)]_{ij}|\right)$ (middle), and $\log_{10}\left(\sum_{f=0:0.01:5}|[(S_y^\diamond(f))^{-1}-(S_x^\diamond(f))^{-1}]_{ij}|\right)$ (right), $i,j\in[120]$, for the MA model, for a single Monte Carlo run: $p=120$ nodes.

We pick $\boldsymbol{\Omega}$ as for the AR model. Only 25% of entries of $\boldsymbol{B}_i^{(q)}$'s are nonzero (randomly picked) and the nonzero elements are independently and uniformly distributed over $[-0.4,-0.2]\cup[0.2,0.4]$. To generate $\boldsymbol{y}$-data, we randomly eliminate one of the 8 clusters ($\boldsymbol{B}_i^{(q)}$'s for randomly picked q) of $\boldsymbol{x}(t)$ and replace it with an independently generated $\boldsymbol{B}_i^{(q)}$, $i\in[3]$, with nonzero entries uniformly distributed over $[-0.2,0.2]$.

For both models, the first 100 samples are discarded to eliminate transients, and we generate $n=n_x=n_y$ observations for $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$, with $n\in\{512,2048,4096\}$. In each run, we calculate the true PSDs $\boldsymbol{S}_x^\diamond(f)$ and $\boldsymbol{S}_y^\diamond(f)$ for $f\in[0,0.5]$ at intervals of 0.01. Let F denote the number of frequencies points in $[0,0.5]$ at intervals of 0.01. Define $\boldsymbol{\Delta}^\diamond(f)=(\boldsymbol{S}_y^\diamond(f))^{-1}-(\boldsymbol{S}_x^\diamond(f))^{-1}$, $b=\max_{i,j\in[p]}(1/F)\sum_f|[(\boldsymbol{S}_x^\diamond(f))^{-1}]_{ij}|$, and $d_{ij}=(1/F)\sum_f|[\boldsymbol{\Delta}^\diamond(f)]_{ij}|$. In each run, we take $\{i,j\}\in\mathcal{E}_\Delta^\diamond$ if $d_{ij}>\tau b$, else $\{i,j\}\notin\mathcal{E}_\Delta^\diamond$, where the threshold $\tau=0.001$ for the MA model and $=0.01$ for the AR model. To avoid very "peaky" inverse PSDs, if $b>50,000$ we redraw the samples till this condition is satisfied: it is needed for the MA models. For a typical realization (run), Figs. 1 and 2 show heatmaps of $\log_{10}\left(\sum_{f=0:0.01:5}|[S^{-1}(f)]_{ij}|\right)$, $i,j\in[120]$, for the AR and MA models, respectively. For the chosen AR model, the percentage of distinct connected edges in the differential graph turn out to be $2.0\pm0.4\%$ and for the MA model, they are $2.0\pm1.0\%$.

Simulation results based on 100 runs are shown in Table 1 where the performance measures are F_1-score and Hamming distance (between the estimated and true edgesets) for efficacy in edge detection, and timing per run as a surrogate for computational complexity. All algorithms were run on a Window 11 Enterprise operating system with processor Intel(R) Core(TM) i7-10700 CPU @2.90 GHz with 32 GB RAM, using MATLAB R2023a. We implemented our three proposed approaches, labeled "DTS-FD, log-sum", "DTS-FD, lasso" and "DTS-FD, SCAD" (DTS stands for dependent time series and FD stands for frequency-domain) using log-sum, lasso and SCAD penalties, respectively. For comparison, we implemented two approaches that assume the data is i.i.d. and they are time-domain approaches based on sample covariances. One of them is based on [6] which minimizes lasso-penalized (1) based on difference of precision matrices (labeled "IID, lasso") and the other follows the recent approach of [30] (labeled "IID, log-sum") and it minimizes log-sum-penalized (1) based on difference of precision matrices. Although the approach of [30] is aimed at multi-attribute graphs, the approach therein applies to our problem by setting the number of attribute to one. For our proposed frequency-domain approaches, we used $M=2,4,5$ ($K=127,255,409$) for the MA model and $M=2,4,6$ ($K=127,255,341$) for the AR model, for $n=512,2048,4096$, respectively.

TABLE 1. F_1 **scores, Hamming distances and timings for the synthetic data examples ($p = 120$), averaged over 100 runs (standard deviation σ in parentheses). "DTS-FD, log-sum", "DTS-FD, lasso" and "DTS-FD, SCAD" are the proposed approaches with log-sum, lasso and SCAD penalties, respectively, "IID, lasso" is the time-domain approach of [6] (also [5]) with lasso penalty, and "IID, log-sum" is the time-domain approach of [30] with log-sum penalty.**

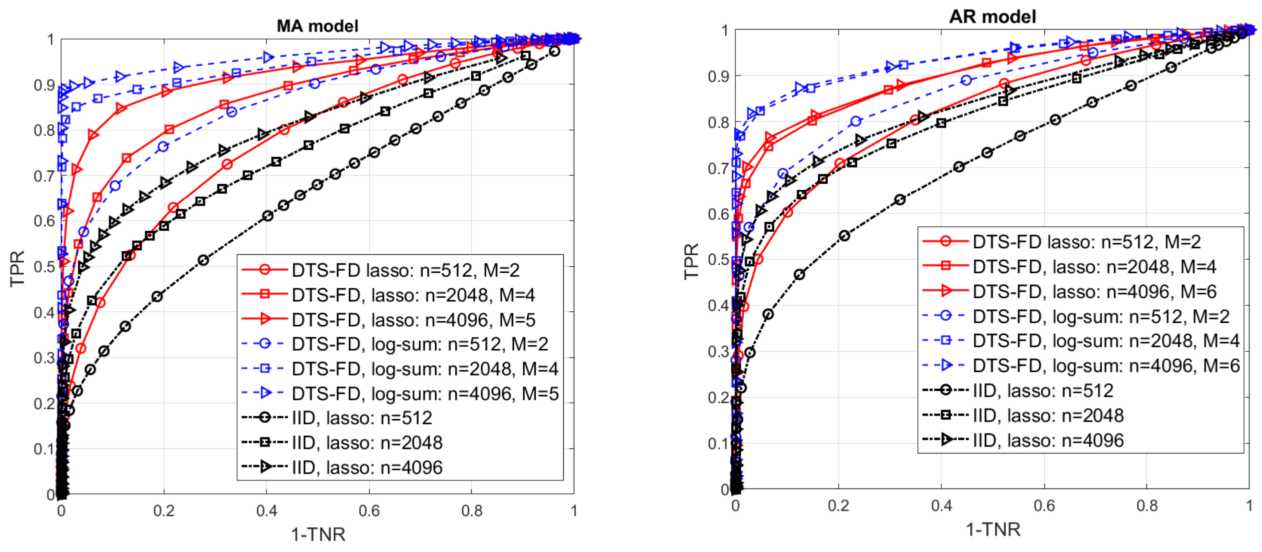
n	512	2048	4096	512	2048	4096
			λ's picked to maximize F_1 score			
	MA model: F_1 score (σ)			AR model: F_1 score (σ)		
IID, lasso [6]	0.21 (0.06)	0.32 (0.09)	0.43 (0.10)	0.25 (0.09)	0.46 (0.10)	0.54 (0.10)
IID, log-sum [30]	0.28 (0.06)	0.46 (0.10)	0.58 (0.11)	0.32 (0.07)	0.53 (0.07)	0.62 (0.09)
DTS-FD, lasso	0.25 (0.09)	0.48 (0.20)	0.63 (0.19)	0.40 (0.12)	0.65 (0.11)	0.69 (0.12)
DTS-FD, log-sum	0.46 (0.10)	0.81 (0.17)	0.91 (0.15)	0.54 (0.10)	0.79 (0.09)	0.82 (0.09)
DTS-FD, SCAD	0.24 (0.08)	0.48 (0.20)	0.68 (0.19)	0.41 (0.12)	0.65 (0.11)	0.69 (0.12)
	MA model: Hamming distance (σ)			AR model: Hamming distance (σ)		
IID, lasso	232.2 (159.4)	220.0 (162.8)	143.0 (97.9)	191.6 (107.0)	157.8 (101.3)	132.1 (80.6)
IID, log-sum	173.2 (62.4)	136.2 (85.1)	105.2 (78.9)	153.3 (32.7)	112.6 (28.2)	94.0 (38.1)
DTS-FD, lasso	373.5 (361.5)	322.2 (392.6)	146.4 (191.1)	203.1 (172.4)	100.0 (76.1)	98.7 (85.4)
DTS-FD, log-sum	128.9 (69.3)	69.4 (88.1)	37.2 (79.1)	125.0 (79.0)	55.6 (31.8)	50.4 (36.3)
DTS-FD, SCAD	386.3 (372.6)	330.5 (408.6)	139.9 (189.5)	203.4 (172.5)	100.2 (75.9)	99.0 (85.3)
	MA model: Timing (s) (σ)			AR model: Timing (s) (σ)		
IID, lasso	0.013 (0.003)	0.010 (0.002)	0.011 (0.002)	0.010 (0.004)	0.008 (0.003)	0.009 (0.003)
IID, log-sum	0.040 (0.005)	0.032 (0.007)	0.030 (0.005)	0.034 (0.007)	0.028 (0.004)	0.026 (0.003)
DTS-FD, lasso	7.6 (0.6)	8.9 (1.4)	11.7 (3.1)	12.6 (3.9)	16.1 (4.2)	17.5 (3.8)
DTS-FD, log-sum	17.4 (2.5)	23.9 (4.0)	33.4 (6.3)	21.2 (5.7)	24.6 (6.2)	28.3 (6.7)
DTS-FD, SCAD	16.5 (1.5)	22.2 (1.9)	26.8 (6.6)	26.2 (7.8)	33.8 (8.9)	35.6 (7.6)
			λ's picked to minimize BIC			
	MA model: F_1 score (σ)			AR model: F_1 score (σ)		
DTS-FD, log-sum	0.47 (0.11)	0.78 (0.17)	0.86 (0.15)	0.51 (0.12)	0.75 (0.10)	0.79 (0.10)
	MA model: Hamming distance (σ)			AR model: Hamming distance (σ)		
DTS-FD, log-sum	160.4 (97.6)	69.9 (85.6)	48.4 (78.3)	183.7 (163.1)	60.9 (30.9)	57.1 (36.8)

It is seen from Table 1 that our log-sum-penalized graph estimator significantly outperforms our lasso based graph estimator which in turn, significantly outperforms the lasso based methods of [5] and [6], yielding higher F_1 scores and lower Hamming distances (ideal F_1 score is 1 and ideal Hamming distance is 0). The performance of our SCAD-penalized graph estimator in Table 1 is similar to that of our lasso based graph estimator showing little improvement. While the log-sum penalized "IID, log-sum" method of [30] improves upon "IID, lasso", it is significantly inferior to our proposed "DTS-FD: log-sum".

The improvement in performance with log-sum penalty over lasso (e.g., "DTS-FD, log-sum" over "DTS-FD, lasso"), and with dependent time-series (DTS-FD) modeling over i.i.d. data modeling (IID) approaches, comes at the cost of much increased computational time. For the MA model, we see from Table 1 that for sample sizes of $n = 512$ and 4096, "DTS-FD, log-sum" approach yields F_1 scores of 0.46 and 0.91, respectively, compared to the "DTS-FD, lasso" F_1 scores of 0.28 and 0.58, respectively, which represent improvements by 84% and 44% (log-sum over lasso), respectively. But the computational cost (timing per run) increases (log-sum over lasso) by factors of 2.29 and 2.28 for $n = 512$ and 4096, respectively. Note that the log-sum solution uses the lasso solution as an initial guess for LLA, and the lasso timing is included in the log-sum timing. One would expect the log-sum solution timing to be approximately twice the lasso timing: solve with lasso, use lasso-based LLA to obtain λ_{ij}'s and solve again using ADMM. On the other hand, we see little improvement in the F_1 score over lasso with the SCAD penalty even though the computational cost for SCAD is comparable to that for log-sum penalty.

For the AR model, we see smaller (compared to the MA model) yet significant improvements in the F_1 scores with log-sum penalty over lasso in Table 1. For sample sizes of $n = 512$ and 4096, "DTS-FD, log-sum" approach yields F_1 scores of 0.54 and 0.82, respectively, compared to the "DTS-FD, lasso" F_1 scores of 0.40 and 0.69, respectively, which represent improvements by 26% and 19% (log-sum over lasso), respectively. The timing per run increases (log-sum over lasso) by factors of 1.68 and 1.63 for $n = 512$ and 4096, respectively.

When analyzing the trade-off between performance and computational time for lasso and log-sum penalties, the Hamming distance performance measure seems to provide a "better" metric. The Hamming distance between the true and the estimated graph edgeset is the sum of the number of distinct incorrect edges in the estimated edgeset (a true edge is missing, or an edge missing from true edgeset is present in the estimated edgeset). For the MA model, for sample sizes of $n = 512$ and 4096, "DTS-FD, lasso" approach yields Hamming distances of 373.5 and 146.4, respectively, compared to the "DTS-FD, log-sum" Hamming distances of 128.9 and 37.2, respectively, which represent reductions by factors of 2.90 and 3.93 (log-sum over lasso), respectively. For the AR model, for sample sizes of $n = 512$ and 4096, "DTS-FD, lasso" approach yields Hamming distances of



(a) ROC curves for the moving average (MA) model.

(b) ROC curves for the autoregressive (AR) model.

FIGURE 3. **ROC curves: "DTS-FD, log-sum" is the proposed approach with log-sum penalty, "DTS-FD, lasso" is the proposed approach with lasso penalty, and "IID, lasso" is the time-domain approach of [6] (also [5]) with lasso penalty. TPR=true positive rate, TNR=true negative rate.**

TABLE 2. **F_1 scores, Hamming distances and timings for AR(3) model with $p \in \{60, 120, 240\}$, averaged over 100 runs (standard deviation σ in parentheses). "DTS-FD, log-sum" and "DTS-FD, lasso" are the proposed approaches with log-sum and lasso penalties, respectively. Also shown is the normalized Hamming distance which is the Hamming distance divided by total number of distinct edges in the differential graph, expressed as percentage.**

n	512	2048	4096	512	2048	4096
	AR model with varying p, λ's picked to maximize F_1 score					
	DTS-FD, lasso: F_1 score (σ)			DTS-FD, log-sum: F_1 score (σ)		
p=60	0.42 (0.16)	0.64 (0.16)	0.74 (0.13)	0.65 (0.12)	0.85 (0.10)	0.89 (0.10)
p=120	0.40 (0.12)	0.65 (0.11)	0.69 (0.12)	0.54 (0.10)	0.79 (0.09)	0.82 (0.09)
p=240	0.39 (0.09)	0.61 (0.07)	0.68 (0.06)	0.40 (0.13)	0.65 (0.07)	0.73 (0.07)
	DTS-FD, lasso: Hamming distance (σ)			DTS-FD, log-sum: Hamming distance (σ)		
p=60	86.1 (75.2)	54.8 (48.6)	31.2 (19.5)	37.4 (15.4)	18.7 (15.3)	14.7 (15.3)
p=120	203.1 (172.4)	100.0 (76.1)	98.7 (85.4)	125.0 (79.0)	55.6 (31.8)	50.4 (36.3)
p=240	734.9 (63.9)	364.7 (97.9)	310.1 (106.6)	1464 (300.8)	374.4 (212.4)	295.5 (120.4)
	DTS-FD, lasso: normalized Hamming distance % (σ)			DTS-FD, log-sum: normalized Hamming distance % (σ)		
p=60	4.71 (4.11)	2.99 (2.66)	1.70 (1.07)	2.04 (0.84)	1.02 (0.84)	0.80 (0.84)
p=120	2.80 (2.37)	1.38 (1.05)	1.36 (1.18)	1.72 (1.09)	0.77 (0.44)	0.69 (0.50)
p=240	2.54 (0.22)	1.26 (0.34)	1.07 (0.37)	5.06 (1.04)	1.29 (0.73)	1.02 (0.42)
	DTS-FD, lasso: Timing (s) (σ)			DTS-FD, log-sum: Timing (s) (σ)		
p=60	3.30 (1.37)	3.40 (0.93)	3.86 (1.03)	4.79 (1.69)	4.87 (1.13)	5.09 (1.10)
p=120	12.6 (3.9)	16.1 (4.2)	17.5 (3.8)	21.2 (5.7)	24.6 (6.2)	28.3 (6.7)
p=240	63.5 (16.1)	103.5 (23.8)	135.4 (27.7)	135.7(44.8)	134.7 (31.2)	142.8 (30.0)

203.1 and 98.7, respectively, compared to the "DTS-FD, log-sum" Hamming distances of 125.0 and 50.4, respectively, which represent reductions by factors of 1.62 and 1.96 (log-sum over lasso), respectively. Thus we have reduction in the Hamming distance by factors of 2.90 and 3.93 with a computational cost increase by factors of 2.29 and 2.28 for $n = 512$ and 4096, respectively, for the MA model, and reduction in the Hamming distance by factors of 1.62 and 1.96 with a computational cost increase by factors of 1.68 and 1.63 for $n = 512$ and 4096, respectively, for the AR model. Since the main objective of differential graph learning is determination of the true edgeset, such a trade-off seems to be reasona

In Table 1, λ's were first picked from a grid of values to maximize the F_1 score (ground truth is known in synthetic data examples) – this establishes how well a method will perform if λ's are judiciously picked. For log-sum penalty we also show the results when λ's are selected to minimize the BIC criterion of Sec. V-B. We see that the heuristic BIC-type criterion performs well.

The receiver operating characteristic (ROC) curves are shown in Fig. 3 for three approaches "DTS-FD, log-sum", "DTS-FD, lasso" and "IID, lasso". By changing the penalty parameter λ and determining the resulting edges over 100 runs, we calculated the true positive rate (TPR) which calculates true edges correctly detected ($\|\hat{\hat{\boldsymbol{\Delta}}}^{(ij)}\| \neq 0$ and $\|(\tilde{\boldsymbol{\Delta}}^{\diamond})^{(ij)}\| \neq 0$), and false positive rate 1-TNR (where TNR is the true negative rate) which are the edges $\{i, j\}$ for which $\|\hat{\hat{\boldsymbol{\Delta}}}^{(ij)}\| \neq 0$ but $\|(\tilde{\boldsymbol{\Delta}}^{\diamond})^{(ij)}\| = 0$. It is seen from Fig. 3 that our log-sum-penalized graph estimator significantly outperforms both the "IID, lasso" approach and our lasso based graph estimator, yielding much higher TPR for a given 1-TNR, consistent with the results of Table 1.

B. GRAPHS WITH VARYING NUMBER OF NODES

We now consider AR(3) models for time-dependent data generation with varying number of graph nodes $p \in \{60, 120, 240\}$. The objective is to empirically study the performance stability of the proposed solutions with varying model dimensions. The AR(3) model follows (90) where $\boldsymbol{A}_i \in \mathbb{R}^{p\times p}$, $i \in [3]$, is block-diagonal with

(i) six 10×10 sub-blocks $\boldsymbol{A}_i^{(q)}$, $q \in [6]$, when $p = 60$,
(ii) eight 15×15 sub-blocks $\boldsymbol{A}_i^{(q)}$, $q \in [8]$, when $p = 120$ (as in Sec. VII-A),
(iii) eight 30×30 sub-blocks $\boldsymbol{A}_i^{(q)}$, $q \in [6]$, when $p = 240$.

All other details regarding generation of $\boldsymbol{\Omega}$, $\boldsymbol{A}_i^{(q)}$'s, $\{\boldsymbol{x}(t)\}$ and $\{\boldsymbol{y}(t)\}$ are exactly as before in Sec. VII-A. The percentage of distinct connected edges in the differential graphs turn out to be $3.0 \pm 1.0\%$, $2.0 \pm 0.4\%$ and $2.0 \pm 0.2\%$ for $p = 60$, $p = 120$ and $p = 240$, respectively.

Simulation results based on 100 runs are shown in Table 2 for the proposed "DTS-FD, lasso" and "DTS-FD, log-sum" approaches where the performance measures, as in Table 1, are the F_1-score, the Hamming distance and timing per run. The results for $p = 120$ are as in Table 1. Since the number of distinct connected edges in the differential graph vary with p, we also show the normalized Hamming distance which is the Hamming distance divided by total number of distinct edges in the differential graph, expressed as percentage. As for Table 1, we used $M = 2, 4, 6$ ($K = 127, 255, 341$) for all AR models, for $n = 512, 2048, 4096$, respectively. The number of unknowns in $\boldsymbol{\Delta}_k$ is p^2, therefore, the number of unknowns being estimated is Mp^2. It is seen in Table 2 that the F_1 score decreases and the Hamming distance increases (i.e., the performance deteriorates) with increasing dimension p for the same sample size n since the number of unknowns being estimated increases. The performance is stable with increasing p as the performance improves with increasing n, and the deterioration in the performance measures with increasing p for fixed n is "gradual."

VIII. REAL DATA: FINANCIAL TIME SERIES

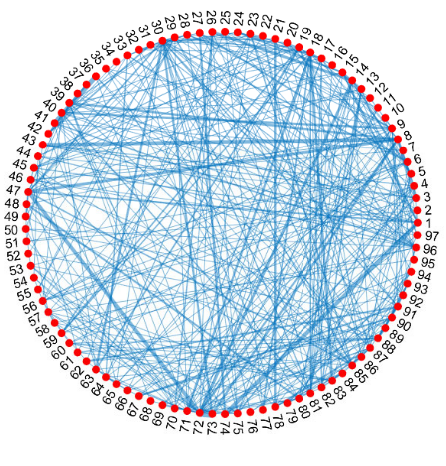
Here we investigate differences in the time series graphical models of the share prices of 97 stocks in the S&P 100 index over two different time periods: Jan. 2, 2013 to Jan. 14, 2015 and Dec. 17, 2015 to Jan. 1, 2018. In the real data example our goal is visualization and exploration of the differential conditional dependency structure underlying the data since the ground truth is unknown. The selection of the duration of each period leads to equal number of samples in the two time periods.

We consider daily share prices (at close of the day) of 97 stocks in the S&P 100 index from Jan. 1, 2013 through Jan. 1, 2018. This data was gathered from Yahoo Finance website. If $z_m(t)$ is share price of mth stock on day t, we pre-process to create $x_m(t) = \ln(z_m(t)/z_m(t-1))$ as the time series to analyze. Such transformations are common in the analysis of financial time series. For instance, such pre-processing is used in [15, Sec. 5.2] for topology selection for graphical models for international stock market data, and in [49, Sec. 5.2] for analyzing GDP growth, total manufacturing production growth and consumer price index core inflation data. We have $x_m(t) = \ln(z_m(t)) - \ln(z_m(t-1))$ which implies that we first perform $\log(\cdot)$ transformation (generally believed to make data "more Gaussian"), followed by lag one differencing to make the data close to univariate uncorrelated and stationary.

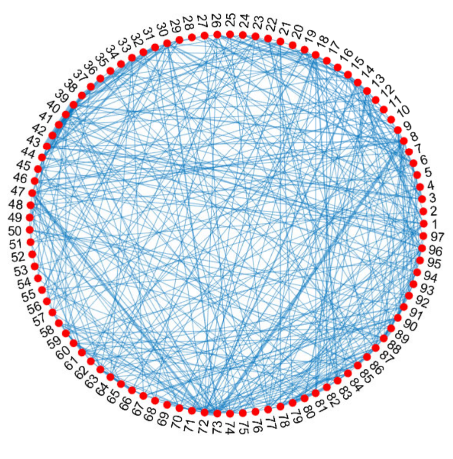
The 97 stocks in the S&P 100 index are classified into 11 sectors (according to the Global Industry Classification Standard (GISC)) and we order the nodes to group them as information technology (nodes 1-12), health care (13)-(27), financials (28)-(44), real estate (45)-(46), consumer discretionary (47)-(56), industrials (57)-(68), communication services (69)-(76), consumer staples (77)-(87), energy (88)-(92), materials (93), utilities (94)-(97). For each m, $x_m(t)$ was centered and normalized to unit variance. The pre-processed data from Jan. 2, 2013 to Jan. 14, 2015 was taken as the $\boldsymbol{x}$-data and that from Dec. 17, 2015 to Jan. 1, 2018 (each series with 512 samples) was taken as the $\boldsymbol{y}$-data. The resulting differential graphs are shown in Fig. 4. The tuning parameter λ as selected as discussed in Sec. V-B. The proposed log-sum penalty yields the sparsest graph with 151 edges. Some of the "strongly" connected nodes (thicker lines and higher degrees) in Fig. 4(d) are Apple (labeled node 1), Meta (72), Alphabet (73), Microsoft (8), Visa (43), Amazon (47), Abbott Labs (14) and American Express (30). The IID model based differential graphs in Figs. 4(a) and 4(b) are just too dense.

IX. CONCLUSION

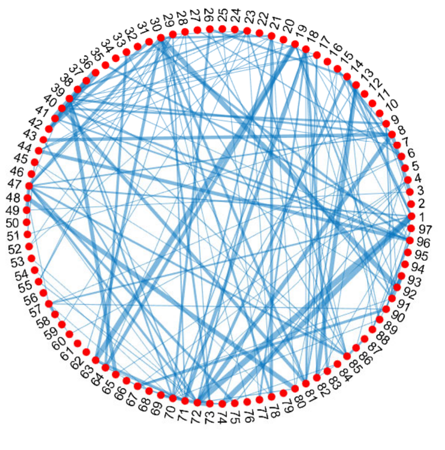
Estimation of differences in CIGs of two TSGGMs was investigated where the two TSGGMs are known to have similar structure. We presented and analyzed a penalized D-trace loss function approach in the frequency domain for differential graph learning using both convex (group lasso) and non-convex (log-sum and SCAD group penalties) regularization functions. An ADMM algorithm was presented to optimize the objective function where, for non-convex penalties, a local linear approximation approach was used. A model selection method for tuning parameter selection was also presented. Both synthetic and real data examples were presented to illustrate the proposed approach where in synthetic data examples, our frequency-domain based log-sum-penalized differential time-series graph estimator



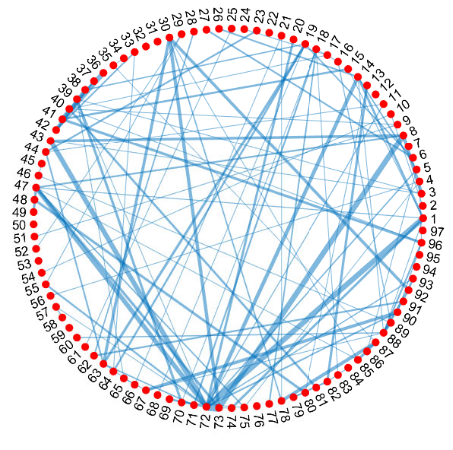
(a) IID, lasso: 429 edges



(b) IID, log-sum: 462 edges



(c) Freq-domain, lasso: 205 edges



(d) Freq-domain, log-sum: 151 edges

FIGURE 4. Differential graphs comparing financial time series (S&P 97 stocks share prices) over period Jan. 2, 2013 to Jan. 14, 2015 with that over period Dec. 17, 2015 to Jan. 1, 2018 (each series with 512 samples): (a) time-domain IID model with lasso penalty [6] (IID, lasso), (b) time-domain IID model with log-sum penalty [30] (IID, log-sum), (c) proposed freq-domain approach with group lasso penalty (FD-DTS, lasso), (d) proposed freq-domain approach with group log-sum penalty (FD-DTS, log-sum). In the freq-domain approaches we used $M = 2$ ($m_t = 63$, $K = 127$). In the figures the thickness of the lines reflects the strength of the connection (determined by $\|\hat{\hat{\Delta}}^{(ij)}\|$).

significantly outperformed our frequency-domain based lasso-penalized differential time-series graph estimator, with F_1 score as the performance metric. Our frequency-domain estimators significantly outperformed the i.i.d. modeling based time domain methods of [5] and [6] (lasso penalty) and [30] (log-sum penalty). The SCAD penalty resulted in little improvement over our lasso based graph estimator.

Theoretical analysis establishing sufficient conditions for consistency and graph recovery was presented using the framework of [20] which however, does not apply to the SCAD penalty. Exploration of alternative analysis techniques (e.g., [47]) to handle penalties such as SCAD, and to analyze (51) instead of its LLA approximation, is of interest.

APPENDIX A
DERIVATION OF (32)

Now we derive (32). By (28)-(30), we have

$$\begin{aligned} \boldsymbol{I}_p \otimes \boldsymbol{I}_p &= (\boldsymbol{Q}_y^* \boldsymbol{Q}_y^T) \otimes (\boldsymbol{Q}_x \boldsymbol{Q}_x^H) \\ &= \big(\boldsymbol{Q}_y^* \otimes \boldsymbol{Q}_x\big)\big(\boldsymbol{Q}_y^T \otimes \boldsymbol{Q}_x^H\big)\big) \\ &= \big(\boldsymbol{Q}_y^* \otimes \boldsymbol{Q}_x\big)\big(\boldsymbol{I}_p \otimes \boldsymbol{I}_p\big)\big(\boldsymbol{Q}_y^T \otimes \boldsymbol{Q}_x^H\big)\big) \end{aligned} \tag{92}$$

and

$$\hat{\boldsymbol{\Sigma}}_y^* \otimes \hat{\boldsymbol{\Sigma}}_x = \big(\boldsymbol{Q}_y^* \otimes \boldsymbol{Q}_x\big)\big(\boldsymbol{D}_y \otimes \boldsymbol{D}_x\big)\big(\boldsymbol{Q}_y^T \otimes \boldsymbol{Q}_x^H\big)\big). \tag{93}$$

Let

$$\boldsymbol{C} = \hat{\boldsymbol{\Sigma}}_x - \hat{\boldsymbol{\Sigma}}_y + \frac{\rho}{2}(\boldsymbol{W}^{(i)} - \boldsymbol{U}^{(i)}). \tag{94}$$

Then by (28) and (31),

$$\begin{aligned} \mathrm{vec}(\boldsymbol{\Delta}) &= \big(\boldsymbol{Q}_y^\top \otimes \boldsymbol{Q}_x^H\big)^{-1} \Big(\boldsymbol{D}_y \otimes \boldsymbol{D}_x + \frac{\rho}{2}\boldsymbol{I}_p \otimes \boldsymbol{I}_p\Big)^{-1} \\ &\quad \times \big(\boldsymbol{Q}_y^* \otimes \boldsymbol{Q}_x\big)^{-1} \mathrm{vec}(\boldsymbol{C}) \\ &= \big(\boldsymbol{Q}_y^* \otimes \boldsymbol{Q}_x\big) \Big(\boldsymbol{D}_y \otimes \boldsymbol{D}_x + \frac{\rho}{2}\boldsymbol{I}_p \otimes \boldsymbol{I}_p\Big)^{-1} \\ &\quad \times \big(\boldsymbol{Q}_y^\top \otimes \boldsymbol{Q}_x^H\big)\mathrm{vec}(\boldsymbol{C}) \\ &= \big(\boldsymbol{Q}_y^* \otimes \boldsymbol{Q}_x\big) \Big(\boldsymbol{D}_y \otimes \boldsymbol{D}_x + \frac{\rho}{2}\boldsymbol{I}_p \otimes \boldsymbol{I}_p\Big)^{-1} \end{aligned}$$

$$\begin{aligned} &\times \mathrm{vec}(\boldsymbol{Q}_x^H \boldsymbol{C} \boldsymbol{Q}_y) \\ &= (\boldsymbol{Q}_y^* \otimes \boldsymbol{Q}_x)\, \mathrm{vec}(\boldsymbol{D} \circ (\boldsymbol{Q}_x^H \boldsymbol{C} \boldsymbol{Q}_y)) \\ &= \mathrm{vec}\Big(\boldsymbol{Q}_x \Big[\boldsymbol{D} \circ (\boldsymbol{Q}_x^H \boldsymbol{C} \boldsymbol{Q}_y)\Big] \boldsymbol{Q}_y^H\Big) . \end{aligned} \tag{95}$$

The desired (32) follows from (94) and (95).

APPENDIX B TECHNICAL LEMMAS AND PROOFS OF THEOREMS 1 AND 2

For theoretical analysis we will use the restricted strong convexity (RSC) based results from [20] which are given therein for real-valued vectors variables. Therefore, we first express our cost $\tilde{L}_f(\tilde{\boldsymbol{\Delta}})$ in terms of $vec(\boldsymbol{\Delta}_k)$, $k \in [M_n]$, and then in terms of $vec(Re(\boldsymbol{\Delta}_k))$ and $vec(Im(\boldsymbol{\Delta}_k))$, before invoking [20].

With $\boldsymbol{\psi}_k := vec(\boldsymbol{\Delta}_k)$ define

$$\boldsymbol{\theta}_k := \begin{bmatrix} Re(\boldsymbol{\psi}_k) \\ Im(\boldsymbol{\psi}_k) \end{bmatrix} \in \mathbb{R}^{2p^2}, \quad \bar{\boldsymbol{\psi}}_k := \begin{bmatrix} \boldsymbol{\psi}_k \\ \boldsymbol{\psi}_k^* \end{bmatrix} \in \mathbb{C}^{2p^2}. \tag{96}$$

Then cost $\tilde{L}(\tilde{\boldsymbol{\Delta}})$ of (47) can be re-expressed in terms of $\bar{\boldsymbol{\psi}}_k$s and $\boldsymbol{\theta}_k$s as

$$\mathcal{L}_c(\tilde{\boldsymbol{\psi}}) = \sum_{k=1}^{M_n} \Big(\frac{1}{2} \bar{\boldsymbol{\psi}}_k^H \bar{\mathcal{H}}_k \bar{\boldsymbol{\psi}}_k - \bar{\boldsymbol{\psi}}_k^H \bar{\boldsymbol{b}}_k \Big) \tag{97}$$

where

$$\tilde{\boldsymbol{\psi}} := \Big[\bar{\boldsymbol{\psi}}_1^\top \cdots \bar{\boldsymbol{\psi}}_{M_n}\Big]^\top \in \mathbb{C}^{2p^2 M_n}, \tag{98}$$

$$\bar{\mathcal{H}}_k := \begin{bmatrix} \hat{\boldsymbol{S}}_{yk}^* \otimes \hat{\boldsymbol{S}}_{xk} & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{S}}_{yk} \otimes \hat{\boldsymbol{S}}_{xk}^* \end{bmatrix} \in \mathbb{C}^{2p^2 M_n \times 2p^2 M_n}, \tag{99}$$

$$\bar{\boldsymbol{b}}_k = \begin{bmatrix} \mathrm{vec}(\hat{\boldsymbol{S}}_{xk} - \hat{\boldsymbol{S}}_{yk}) \\ \mathrm{vec}((\hat{\boldsymbol{S}}_{xk} - \hat{\boldsymbol{S}}_{yk})^*) \end{bmatrix} \in \mathbb{C}^{2p^2 M_n}, \tag{100}$$

and

$$\mathcal{L}_r(\tilde{\boldsymbol{\theta}}) = \sum_{k=1}^{M_n} \Big(\boldsymbol{\theta}_k^H \mathcal{H}_k \boldsymbol{\theta}_k - 2\, \boldsymbol{\theta}_k^\top \boldsymbol{b}_k \Big) \tag{101}$$

where ($\iota = \sqrt{-1}$)

$$\tilde{\boldsymbol{\theta}} := \big[\boldsymbol{\theta}_1^\top \cdots \boldsymbol{\theta}_{M_n}^\top\big]^\top \in \mathbb{R}^{2p^2 M_n}, \tag{102}$$

$$\mathcal{H}_k := \frac{1}{2} \boldsymbol{T}_{rc}^H \bar{\mathcal{H}}_k \boldsymbol{T}_{rc} \in \mathbb{R}^{2p^2 M_n \times 2p^2 M_n}, \tag{103}$$

$$\boldsymbol{b}_k := \frac{1}{2} \boldsymbol{T}_{rc}^H \bar{\boldsymbol{b}}_k \in \mathbb{R}^{2p^2 M_n}, \quad \tilde{\boldsymbol{T}}_{rc} := \begin{bmatrix} 1 & \iota \\ 1 & -\iota \end{bmatrix}, \tag{104}$$

$$\boldsymbol{T}_{rc} := \tilde{\boldsymbol{T}}_{rc} \otimes \boldsymbol{I}_{p^2} \in \mathbb{C}^{2p^2}, \quad \bar{\boldsymbol{\psi}}_k = \boldsymbol{T}_{rc} \boldsymbol{\theta}_k, \tag{105}$$

and $\boldsymbol{T}_{rc}$ yields real-to-complex transformation [16, Appendix 2]. Note that we have the equalities $\tilde{L}(\tilde{\boldsymbol{\Delta}}) = \mathcal{L}_c(\tilde{\boldsymbol{\psi}}) = \mathcal{L}_r(\tilde{\boldsymbol{\theta}})$. It is easy to establish that $\|\tilde{\boldsymbol{T}}_{rc}\| = \|\boldsymbol{T}_{rc}\| = \sqrt{2}$ and $\|\tilde{\boldsymbol{T}}_{rc}\|_{1,\infty} = \|\boldsymbol{T}_{rc}\|_{1,\infty} = \|\boldsymbol{T}_{rc}^H\|_{1,\infty} = \|\tilde{\boldsymbol{T}}_{rc}^H\|_{1,\infty} = 2$.

We now turn our attention to the penalty/regularization term $\sum_{i,j=1}^p \lambda_{ij} \|\tilde{\boldsymbol{\Delta}}^{(ij)}\|$ in (57) and will express it to conform to the framework of [20]. Note that the term $\tilde{\boldsymbol{\Delta}}^{(ij)}$ corresponds to the edge $\{i,j\}$ of the graph. We denote its real-valued version as

$$\tilde{\boldsymbol{\theta}}_{Gt} = \Big[Re(\tilde{\boldsymbol{\Delta}}^{(ij)})^\top \;\; Im(\tilde{\boldsymbol{\Delta}}^{(ij)})^\top\Big]^\top \in \mathbb{R}^{2M_n} \tag{106}$$

(subscript G for grouped variables [20]), with index $t \in [p^2]$, $(i,j) \leftrightarrow t = (i-1)p + j$ and $i = \lfloor t/p \rfloor + 1$, $j = t \mod p$. Using this notation, we have (we now denote λ by λ_n)

$$\sum_{i,j=1}^{p} \lambda_{ij} \|\tilde{\boldsymbol{\Delta}}^{(ij)}\| = \lambda_n \sum_{t=1}^{p^2} w_t \|\tilde{\boldsymbol{\theta}}_{Gt}\|_2 , \tag{107}$$

$$w_t = \begin{cases} 1 & : \text{lasso} \\ \epsilon/(\epsilon + \|\bar{\tilde{\boldsymbol{\theta}}}_{Gt}\|) & : \text{log-sum}, \end{cases} \tag{108}$$

where $\bar{\tilde{\boldsymbol{\theta}}}_{Gt}$ corresponds to $\bar{\tilde{\boldsymbol{\Delta}}}^{(ij)}$ In the notation of [20], the regularization penalty without λ_n is expressed as a weighted group norm

$$\mathcal{R}(\tilde{\boldsymbol{\theta}}) = \|\tilde{\boldsymbol{\theta}}\|_{\bar{\mathcal{G}},2w} := \sum_{t=1}^{p^2} w_t \|\tilde{\boldsymbol{\theta}}_{Gt}\|_2 \tag{109}$$

where the index set $\{1, 2, \cdots, 2M_n p^2\}$ is partitioned into a set of $N_G = p^2$ disjoint groups $\bar{\mathcal{G}} = \{G_1, G_2, \cdots, G_{p^2}\}$ and the subscript $2w$ signifies the weighted group norm. Using this notation, the penalized counterpart to $\tilde{L}_f(\tilde{\boldsymbol{\Delta}})$ of (57) is

$$\bar{\mathcal{L}}_r(\tilde{\boldsymbol{\theta}}) = \mathcal{L}_r(\tilde{\boldsymbol{\theta}}) + \lambda_n \mathcal{R}(\tilde{\boldsymbol{\theta}}) . \tag{110}$$

As discussed in [20, Sec. 2.2], w.r.t. the usual Euclidean inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^\top \boldsymbol{v}$ for $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{2M_n p^2}$ and given any subset $S_{\bar{\mathcal{G}}} \subseteq \{1, 2, \cdots, N_G\}$ of group indices, define the subspace

$$\mathcal{M} = \{\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{2M_n p^2} \,|\, \tilde{\boldsymbol{\theta}}_{Gt} = \boldsymbol{0} \text{ for all } t \notin S_{\bar{\mathcal{G}}}\} \tag{111}$$

and its orthogonal complement

$$\mathcal{M}^\perp = \{\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{2M_n p^2} \,|\, \tilde{\boldsymbol{\theta}}_{Gt} = \boldsymbol{0} \text{ for all } t \in S_{\bar{\mathcal{G}}}\} . \tag{112}$$

The chosen $\mathcal{R}(\tilde{\boldsymbol{\theta}})$ is decomposable w.r.t. $(\mathcal{M}, \mathcal{M}^\perp)$ since $\mathcal{R}(\tilde{\boldsymbol{\theta}}^{(1)} + \tilde{\boldsymbol{\theta}}^{(2)}) = \mathcal{R}(\tilde{\boldsymbol{\theta}}^{(1)}) + \mathcal{R}(\tilde{\boldsymbol{\theta}}^{(2)})$ for any $\tilde{\boldsymbol{\theta}}^{(1)} \in \mathcal{M}$ and $\tilde{\boldsymbol{\theta}}^{(2)} \in \mathcal{M}^\perp$ [20, Sec. 2.2,Example 2].

In order to invoke [20], we need the dual norm $\mathcal{R}^{\circledast}$ of regularizer $\mathcal{R}$ w.r.t. the inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^\top \boldsymbol{v}$ (we use $\circledast$ instead of $*$ since $*$ has already been used to denote complex conjugation). It is given by [20, Sec. 2.3]

$$\begin{aligned} \mathcal{R}^{\circledast}(\boldsymbol{v}) &= \sup_{\|\boldsymbol{u}\|_{\bar{\mathcal{G}},2w} \le 1} \langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sup_{\|\boldsymbol{u}\|_{\bar{\mathcal{G}},2w} \le 1} \sum_{i=1}^{2M_n p^2} u_i v_i \\ &\le \sup_{\|\boldsymbol{u}\|_{\bar{\mathcal{G}},2w} \le 1} \sum_{t=1}^{p^2} \|\boldsymbol{u}_{Gt}\|_2 \|\boldsymbol{v}_{Gt}\|_2 \\ &\le \sup_{\|\boldsymbol{u}\|_{\bar{\mathcal{G}},2w} \le 1} \Big(\max_{t \in [p^2]} w_t^{-1} \|\boldsymbol{v}_{Gt}\|_2 \Big) \underbrace{\sum_{t=1}^{p^2} w_t \|\boldsymbol{u}_{Gt}\|_2}_{=\|\boldsymbol{u}\|_{\bar{\mathcal{G}},2w}} \\ &\le \max_{t \in [p^2]} w_t^{-1} \|\boldsymbol{v}_{Gt}\|_2 . \end{aligned} \tag{113}$$

We also need the subspace compatibility index [20], defined as

$$\Psi(\mathcal{M}) = \sup_{\boldsymbol{u}\in\mathcal{M}\setminus\{0\}} \mathcal{R}(\boldsymbol{u})/\|\boldsymbol{u}\|_2 . \tag{114}$$

We have $\mathcal{R}(\boldsymbol{u}) = \sum_{t=1}^{p^2} w_t \|\boldsymbol{u}_{Gt}\|_2 \le (\max_{t\in[p^2]} w_t) \sum_{t=1}^{p^2} \|\boldsymbol{u}_{Gt}\|_2$. By (108), $w_t \le 1$ and by the Cauchy-Schwarz inequality, for $\boldsymbol{u} \in \mathcal{M}$, $\sum_{t=1}^{p^2} \|\boldsymbol{u}_{Gt}\|_2 \le \sqrt{s_n}\,\|\boldsymbol{u}\|_2$. Thus, for the lasso and log-sum penalties, $\Psi(\mathcal{M}) \le \sqrt{s_n}$.

We need to establish a restricted strong convexity condition [20] on $\mathcal{L}_r(\tilde{\boldsymbol{\theta}})$. With $\tilde{\boldsymbol{\theta}}^\diamond$ denoting the true value of $\tilde{\boldsymbol{\theta}}$, let $\tilde{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}^\diamond + \tilde{\boldsymbol{\gamma}}$ with $\boldsymbol{\theta}_k = \boldsymbol{\theta}_k^\diamond + \boldsymbol{\gamma}_k$ (cf. (102)). Consider

$$\delta\mathcal{L}_r(\tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\theta}}^\diamond) := \mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond + \tilde{\boldsymbol{\gamma}}) - \mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond) - \langle \nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond), \tilde{\boldsymbol{\gamma}}\rangle \tag{115}$$

where the gradient $\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond)$ at $\tilde{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}^\diamond$ is

$$\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond) = \Big[(\nabla_1\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))^\top \cdots (\nabla_{M_n}\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))^\top\Big]^\top , \tag{116}$$

$$\nabla_k\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond) := \frac{\partial \mathcal{L}_r(\tilde{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_k}\Big|_{\tilde{\boldsymbol{\theta}}=\tilde{\boldsymbol{\theta}}^\diamond} = 2\mathcal{H}_k\boldsymbol{\theta}_k^\diamond - 2\,\boldsymbol{b}_k . \tag{117}$$

Noting that $\mathcal{H}_k = \mathcal{H}_k^\top$, (115) simplifies to

$$\delta\mathcal{L}_r(\tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\theta}}^\diamond) = \sum_{k=1}^{M_n} \boldsymbol{\gamma}_k^\top \mathcal{H}_k \boldsymbol{\gamma}_k , \tag{118}$$

which may be rewritten as

$$\delta\mathcal{L}_r(\tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\theta}}^\diamond) = \sum_{k=1}^{M_n} \Big[\boldsymbol{\gamma}_k^\top \mathcal{H}_k^\diamond \boldsymbol{\gamma}_k + \boldsymbol{\gamma}_k^\top (\mathcal{H}_k - \mathcal{H}_k^\diamond)\boldsymbol{\gamma}_k\Big]. \tag{119}$$

Under the sparsity assumption (72), $\tilde{\boldsymbol{\theta}}^\diamond = \tilde{\boldsymbol{\theta}}^\diamond_{\mathcal{M}}$, hence, $\tilde{\boldsymbol{\theta}}^\diamond_{\mathcal{M}^\perp} = \boldsymbol{0}$, where $\tilde{\boldsymbol{\theta}}_{\mathcal{M}}$ and $\tilde{\boldsymbol{\theta}}_{\mathcal{M}^\perp}$ denote projection of $\tilde{\boldsymbol{\theta}}$ on subspaces $\mathcal{M}$ and $\mathcal{M}^\perp$, respectively. Similar to $\hat{\tilde{\boldsymbol{\Delta}}} = \arg\min_{\tilde{\boldsymbol{\Delta}}} \tilde{L}_f(\tilde{\boldsymbol{\Delta}})$, suppose

$$\hat{\tilde{\boldsymbol{\theta}}} = \arg\min_{\tilde{\boldsymbol{\theta}}} \big\{\mathcal{L}_r(\tilde{\boldsymbol{\theta}}) + \lambda_n \mathcal{R}(\tilde{\boldsymbol{\theta}})\big\} , \tag{120}$$

and we consider (115) and (118) with $\hat{\tilde{\boldsymbol{\theta}}} = \tilde{\boldsymbol{\theta}}^\diamond + \tilde{\boldsymbol{\gamma}}$. Then

$$\hat{\tilde{\boldsymbol{\theta}}} - \tilde{\boldsymbol{\theta}}^\diamond = \hat{\tilde{\boldsymbol{\theta}}}_{\mathcal{M}} - \tilde{\boldsymbol{\theta}}^\diamond + \hat{\tilde{\boldsymbol{\theta}}}_{\mathcal{M}^\perp} = \tilde{\boldsymbol{\gamma}}_{\mathcal{M}} + \tilde{\boldsymbol{\gamma}}_{\mathcal{M}^\perp} . \tag{121}$$

By [20, Lemma 1],

$$\mathcal{R}(\tilde{\boldsymbol{\gamma}}_{\mathcal{M}^\perp}) \le 3\mathcal{R}(\tilde{\boldsymbol{\gamma}}_{\mathcal{M}}) + 4\mathcal{R}(\tilde{\boldsymbol{\theta}}^\diamond_{\mathcal{M}^\perp}) , \tag{122}$$

if we pick

$$\lambda_n \ge 2\, \mathcal{R}^{\circledast}(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond)) . \tag{123}$$

Since in our case $\tilde{\boldsymbol{\theta}}^\diamond_{\mathcal{M}^\perp} = \boldsymbol{0}$, we have $\mathcal{R}(\tilde{\boldsymbol{\theta}}^\diamond_{\mathcal{M}^\perp}) = 0$.

We now turn to bounding $\mathcal{R}^{\circledast}(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))$. First we need several auxiliary results. Define

$$\boldsymbol{\Delta}_{xk} := \hat{\boldsymbol{S}}_{xk} - \boldsymbol{S}_{xk}^\diamond , \quad \boldsymbol{\Delta}_{yk} := \hat{\boldsymbol{S}}_{yk} - \boldsymbol{S}_{yk}^\diamond , \tag{124}$$

$$\boldsymbol{\Delta}_{yxk} := \hat{\boldsymbol{S}}_{yk}^* \otimes \hat{\boldsymbol{S}}_{xk} - (\boldsymbol{S}_{yk}^\diamond)^* \otimes \boldsymbol{S}_{xk}^\diamond , \tag{125}$$

$$\bar{\delta}_x = \max_{k\in M_n} \|\boldsymbol{\Delta}_{xk}\|_\infty , \quad \bar{\delta}_y = \max_{k\in M_n} \|\boldsymbol{\Delta}_{yk}\|_\infty , \tag{126}$$

$$\bar{\delta} \ge \max\{\bar{\delta}_x, \bar{\delta}_y\} . \tag{127}$$

Lemma 1: Under (124)-(127) and with B_{xy} as in (73), we have

$$\|\boldsymbol{\Delta}_{yxk}\|_\infty \le \bar{\delta}^2 + 2\,B_{xy}\,\bar{\delta} =: \bar{B} . \tag{128}$$

Proof: We can rewrite $\boldsymbol{\Delta}_{yxk}$ as

$$\boldsymbol{\Delta}_{yxk} = \boldsymbol{\Delta}_{yk}^* \otimes \boldsymbol{\Delta}_{xk} + (\boldsymbol{S}_{yk}^\diamond)^* \otimes \boldsymbol{\Delta}_{xk} + \boldsymbol{\Delta}_{yk}^* \otimes \boldsymbol{S}_{xk}^\diamond . \tag{129}$$

Therefore

$$\begin{aligned}\|\boldsymbol{\Delta}_{yxk}\|_\infty &\le \|\boldsymbol{\Delta}_{yk}\|_\infty \|\boldsymbol{\Delta}_{xk}\|_\infty + \|(\boldsymbol{S}_{yk}^\diamond\|_\infty \|\boldsymbol{\Delta}_{xk}\|_\infty \\ &\quad + \|\boldsymbol{\Delta}_{yk}\|_\infty \|\boldsymbol{S}_{xk}^\diamond\|_\infty \\ &\le \bar{\delta}_y\bar{\delta}_x + B_{xy}\bar{\delta}_x + \bar{\delta}_y B_{xy} \le \bar{\delta}^2 + 2B_{xy}\bar{\delta} . \quad \blacksquare\end{aligned} \tag{130}$$

Using the notation G_t for the group t corresponding to the edge $\{i,j\}$, as in (106), let $(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt} \in \mathbb{R}^{2M}$ denote the corresponding entries of the gradient. By (116)-(117), we have

$$(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt} = \Big[(\nabla_1\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}^\top \cdots (\nabla_{M_n}\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}^\top\Big]^\top , \tag{131}$$

$$(\nabla_k\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt} = (2\mathcal{H}_k\boldsymbol{\theta}_k^\diamond - 2\,\boldsymbol{b}_k)_{Gt} \in \mathbb{R}^2 . \tag{132}$$

At the true values $\mathcal{H}_k = \mathcal{H}_k^\diamond$ and $\boldsymbol{b}_k = \boldsymbol{b}_k^\diamond$,

$$\nabla_k\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond)\Big|_{\mathcal{H}_k=\mathcal{H}_k^\diamond, \boldsymbol{b}_k=\boldsymbol{b}_k^\diamond} = \boldsymbol{0} = 2\mathcal{H}_k^\diamond\boldsymbol{\theta}_k^\diamond - 2\,\boldsymbol{b}_k^\diamond \tag{133}$$

(cf. (14)-(15)) where

$$\mathcal{H}_k^\diamond := \frac{1}{2}\boldsymbol{T}_{rc}^H \bar{\mathcal{H}}_k^\diamond \boldsymbol{T}_{rc} , \quad \boldsymbol{b}_k^\diamond := \frac{1}{2}\boldsymbol{T}_{rc}^H \bar{\boldsymbol{b}}_k^\diamond \tag{134}$$

$$\bar{\mathcal{H}}_k^\diamond := \begin{bmatrix} (\boldsymbol{S}_{yk}^\diamond)^* \otimes \boldsymbol{S}_{xk}^\diamond & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_{yk}^\diamond \otimes (\boldsymbol{S}_{xk}^\diamond)^* \end{bmatrix} , \tag{135}$$

$$\bar{\boldsymbol{b}}_k^\diamond = \begin{bmatrix} \mathrm{vec}(\boldsymbol{S}_{xk}^\diamond - \boldsymbol{S}_{yk}^\diamond) \\ \mathrm{vec}((\boldsymbol{S}_{xk}^\diamond - \boldsymbol{S}_{yk}^\diamond)^*) \end{bmatrix} . \tag{136}$$

Therefore, we may rewrite (132) as

$$\begin{aligned}(\nabla_k\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt} &= (2(\mathcal{H}_k - \mathcal{H}_k^\diamond)\boldsymbol{\theta}_k^\diamond - 2\,(\boldsymbol{b}_k - \boldsymbol{b}_k^\diamond))_{Gt} \\ &= \sum_{q=1}^{p^2} \Big[2(\mathcal{H}_k - \mathcal{H}_k^\diamond)_{Gt,Gq}(\boldsymbol{\theta}_k^\diamond)_{Gq}\Big] - 2\,(\boldsymbol{b}_k - \boldsymbol{b}_k^\diamond)_{Gt}\end{aligned} \tag{137}$$

where G_q represents group q corresponding to some edge $\{\ell, m\}$, $(i,j) \leftrightarrow t = (i-1)p+j$ and $(\ell, m) \leftrightarrow q = (\ell-1)p+m$.

Lemma 2: Under the conditions of Lemma 1

$$\|(\nabla_k\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}\|_2 \le \sum_{q=1}^{p^2} 2\,\bar{B}\,\|(\boldsymbol{\theta}_k^\diamond)_{Gq}\|_2 + 4\bar{\delta} . \tag{138}$$

Proof: With $(i,j) \leftrightarrow t = (i-1)p+j$ and $(\ell, m) \leftrightarrow q = (\ell-1)p+m$, we have

$$(\bar{\mathcal{H}}_k - \bar{\mathcal{H}}_k^\diamond)_{Gt,Gq} = \begin{bmatrix} a_k & 0 \\ 0 & a_k^* \end{bmatrix} , \tag{139}$$

$$a_k := [\hat{\boldsymbol{S}}_{yk}^*]_{jm}[\hat{\boldsymbol{S}}_{xk}]_{i\ell} - [(\boldsymbol{S}_{yk}^\diamond)^*]_{jm}[\boldsymbol{S}_{xk}^\diamond]_{i\ell} , \tag{140}$$

$$(\bar{\boldsymbol{b}}_k - \bar{\boldsymbol{b}}_k^\diamond))_{Gt} = \begin{bmatrix} [\boldsymbol{\Delta}_{xk} - \boldsymbol{\Delta}_{yk}]_{ij} \\ [\boldsymbol{\Delta}_{xk} - \boldsymbol{\Delta}_{yk}]_{ij}^* \end{bmatrix}, \tag{141}$$

where (139)-(140) follow from $\text{vec}(\boldsymbol{S}_{xk}\boldsymbol{\Delta}_k\boldsymbol{S}_{yk}) = (\boldsymbol{S}_{yk}^\top \otimes \boldsymbol{S}_{xk})\text{vec}(\boldsymbol{\Delta}_k)$ and $\boldsymbol{S}_{yk}^\top = \boldsymbol{S}_{yk}^*$. Using Lemma 1, $\|\tilde{\boldsymbol{T}}_{rc}\| = \sqrt{2}$ and $\|(\bar{\mathcal{H}}_k - \bar{\mathcal{H}}_k^\diamond)_{Gt,Gq}\| \leq \|\boldsymbol{\Delta}_{yxk}\|_\infty$, we have

$$\begin{aligned}\|2(\mathcal{H}_k - \mathcal{H}_k^\diamond)_{Gt,Gq}\| &= \|\tilde{\boldsymbol{T}}_{rc}^H(\bar{\mathcal{H}}_k - \bar{\mathcal{H}}_k^\diamond)_{Gt,Gq}\tilde{\boldsymbol{T}}_{rc}\| \\ &\leq \|\tilde{\boldsymbol{T}}_{rc}^H\|\; \|(\bar{\mathcal{H}}_k - \bar{\mathcal{H}}_k^\diamond)_{Gt,Gq}\|\; \|\tilde{\boldsymbol{T}}_{rc}\| \leq 2\,\bar{B}\,. \end{aligned} \tag{142}$$

By (124), (126), (127) and (141)

$$\begin{aligned}\|2\,(\boldsymbol{b}_k - \boldsymbol{b}_k^\diamond)_{Gt}\|_2 &= \|\tilde{\boldsymbol{T}}_{rc}^H(\bar{\boldsymbol{b}}_k - \bar{\boldsymbol{b}}_k^\diamond))_{Gt}\| \\ &\leq \|\tilde{\boldsymbol{T}}_{rc}^H\|\; \|(\bar{\boldsymbol{b}}_k - \bar{\boldsymbol{b}}_k^\diamond))_{Gt}\|_2 \leq 4\bar{\delta}\,. \end{aligned} \tag{143}$$

By (137), (142) and (143) we have (138). ■

Lemma 3: Under the conditions of Lemma 1 if $\bar{\delta} \leq B_{xy}$,

$$\mathcal{R}^\circledast(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond)) \leq B_{\text{init}}\sqrt{M_n}\,\left(6\,B_{xy}B_d s_n + 4\right)\,\bar{\delta} \tag{144}$$

where B_d and B_{init} are given by (74) and (81), respectively.

Proof: By Lemma 2 and (131),

$$\begin{aligned}\|(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}\|_2 &= \sqrt{\sum_{k=1}^{M_n}\|(\nabla_k\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}\|_2^2} \\ &\leq \sqrt{M_n}\max_{k\in[M_n]}\|(\nabla_k\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}\|_2 \\ &\leq \sqrt{M_n}\Big[2\,\bar{B}\max_{k\in[M_n]}\Big(\sum_{q=1}^{p^2}\|(\boldsymbol{\theta}_k^\diamond)_{Gq}\|_2\Big) + 4\bar{\delta}\Big]. \end{aligned} \tag{145}$$

Observe that $\sum_{q=1}^{p^2}\|(\boldsymbol{\theta}_k^\diamond)_{Gq}\|_2 \leq s_n \max_{q\in[p^2]}\|(\boldsymbol{\theta}_k^\diamond)_{Gq}\|_2$ since at most s_n edges are connected in the true graph. For group q with $(\ell, m) \leftrightarrow q = (\ell-1)p + m$, $\|(\boldsymbol{\theta}_k^\diamond)_{Gq}\|_2 = |[\boldsymbol{\Delta}_k^\diamond]_{\ell m}| \leq B_d$ for $k \in [M_n]$. Therefore,

$$\|(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}\|_2 \leq \sqrt{M_n}\Big[2\,\bar{B}s_nB_d + 4\bar{\delta}\Big]. \tag{146}$$

By (113) and (146)

$$\begin{aligned}\mathcal{R}^\circledast(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond)) &\leq \max_{t\in[p^2]} w_t^{-1}\|(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}\|_2 \\ &\leq (\max_{t\in[p^2]} w_t^{-1})\,(\max_{t\in[p^2]}\|(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))_{Gt}\|_2) \\ &\overset{\bar{\delta}\leq B_{xy}}{\leq} B_{\text{init}}\sqrt{M_n}\,\left(6\,B_{xy}B_d s_n + 4\right)\,\bar{\delta} \end{aligned} \tag{147}$$

where, for the log-sum penalty we used $\max_{t\in[p^2]} w_t^{-1} = 1 + \max_{t\in[p^2]}\|\tilde{\boldsymbol{\theta}}_{Gt}\|/\epsilon = 1 + \max_{i,j\in[p]}\|\bar{\tilde{\boldsymbol{\Delta}}}^{(ij)}\|/\epsilon =: B_{\text{init}}$, and $\bar{\delta} \leq B_{xy}$ results in $\bar{B} = \bar{\delta}^2 + 2\,B_{xy} \leq 3B_{xy}\bar{\delta}$. ■

Lemma 4: Under the conditions of Lemmas 1 and 3, if $\lambda_n \geq 2\mathcal{R}^\circledast(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))$,

$$\delta\mathcal{L}_r(\tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\theta}}^\diamond) \geq \kappa_{\mathcal{L}}\,\|\tilde{\boldsymbol{\gamma}}\|_2^2\,, \tag{148}$$

where $\kappa_{\mathcal{L}} = \phi_{\min}^\diamond - 192\,s_nM_nB_{\text{init}}^2B_{xy}\bar{\delta}$.

Proof: Consider (119). By (103) we have

$$\begin{aligned}\sum_{k=1}^{M_n}\boldsymbol{\gamma}_k^\top\mathcal{H}_k^\diamond\boldsymbol{\gamma}_k &= \sum_{k=1}^{M_n}\frac{1}{2}(\boldsymbol{T}_{rc}\boldsymbol{\gamma}_k)^H\bar{\mathcal{H}}_k^\diamond(\boldsymbol{T}_{rc}\boldsymbol{\gamma}_k) \\ &\geq \sum_{k=1}^{M_n}\frac{1}{2}\phi_{\min}(\bar{\mathcal{H}}_k^\diamond)\|\boldsymbol{T}_{rc}\boldsymbol{\gamma}_k\|_2^2 \\ &= \sum_{k=1}^{M_n}\phi_{\min}(\bar{\mathcal{H}}_k^\diamond)\|\boldsymbol{\gamma}_k\|_2^2 \;\text{ since }\; \boldsymbol{T}_{rc}^H\boldsymbol{T}_{rc} = 2\boldsymbol{I}_{p^2}\,. \end{aligned} \tag{149}$$

Now $\phi_{\min}(\bar{\mathcal{H}}_k^\diamond) = \phi_{\min}(\boldsymbol{S}_{yk}^\diamond)\phi_{\min}(\boldsymbol{S}_{xk}^\diamond) \geq \phi_{\min}^\diamond$, implying

$$\sum_{k=1}^{M_n}\boldsymbol{\gamma}_k^\top\mathcal{H}_k^\diamond\boldsymbol{\gamma}_k \geq \phi_{\min}^\diamond\sum_{k=1}^{M_n}\|\boldsymbol{\gamma}_k\|_2^2 = \phi_{\min}^\diamond\|\tilde{\boldsymbol{\gamma}}\|_2^2\,. \tag{150}$$

Define

$$\check{\mathcal{H}} := \text{block-diag}\big\{\mathcal{H}_1\,,\;\cdots\,,\;\mathcal{H}_{M_n}\big\}\,, \tag{151}$$

$$\check{\mathcal{H}}^\diamond := \text{block-diag}\big\{\mathcal{H}_1^\diamond\,,\;\cdots\,,\;\mathcal{H}_{M_n}^\diamond\big\}\,. \tag{152}$$

We have $\|\check{\mathcal{H}} - \check{\mathcal{H}}^\diamond\|_\infty = \max_{k\in[M_n]}\|\mathcal{H}_k - \mathcal{H}_k^\diamond\|_\infty$. Using the facts $\|\boldsymbol{AB}\|_\infty \leq \|\boldsymbol{A}\|_\infty\,\|\boldsymbol{B}^\top\|_{1,\infty}$ and $\|\boldsymbol{AB}\|_\infty \leq \|\boldsymbol{B}\|_\infty\,\|\boldsymbol{A}\|_{1,\infty}$, and Lemma 1, we have

$$\begin{aligned}\|\mathcal{H}_k - \mathcal{H}_k^\diamond\|_\infty &= \|\frac{1}{2}\boldsymbol{T}_{rc}^H(\bar{\mathcal{H}}_k - \bar{\mathcal{H}}_k^\diamond)\boldsymbol{T}_{rc}\|_\infty \\ &\leq \frac{1}{2}\|\boldsymbol{T}_{rc}^\top\|_{1,\infty}^2\|\bar{\mathcal{H}}_k - \bar{\mathcal{H}}_k^\diamond\|_\infty \leq \frac{4}{2}\|\boldsymbol{\Delta}_{yxk}\|_\infty \leq 2\bar{B}\,. \end{aligned} \tag{153}$$

By (151)-(153),

$$\begin{aligned}\Big|\sum_{k=1}^{M_n}\boldsymbol{\gamma}_k^\top(\mathcal{H}_k - \mathcal{H}_k^\diamond)\boldsymbol{\gamma}_k\Big| &= |\tilde{\boldsymbol{\gamma}}^\top(\check{\mathcal{H}} - \check{\mathcal{H}}^\diamond)\tilde{\boldsymbol{\gamma}}| \\ &\leq \sum_{\ell=1}^{2p^2M_n}\sum_{m=1}^{2p^2M_n}|\tilde{\gamma}_\ell[\check{\mathcal{H}} - \check{\mathcal{H}}^\diamond]_{\ell m}\tilde{\gamma}_m| \\ &\leq \|\check{\mathcal{H}} - \check{\mathcal{H}}^\diamond\|_\infty\Big(\sum_{m=1}^{2p^2M_n}|\tilde{\gamma}_m|\Big)^2 =: A\,. \end{aligned} \tag{154}$$

As in (102), $\tilde{\boldsymbol{\gamma}} = \big[\boldsymbol{\gamma}_1^\top,\;\cdots\,,\;\boldsymbol{\gamma}^\top\big]^\top$. Expressing in terms of group G_t and using the Cauchy-Schwarz inequality, we have

$$\begin{aligned}\sum_{m=1}^{2p^2M_n}|\tilde{\gamma}_m| &= \sum_{t=1}^{p^2}\Big(\sum_{k=1}^{M_n}\big[|\gamma_{kt}| + |\gamma_{k(t+p^2)}|\big]\Big) \\ &\leq \sum_{t=1}^{p^2}\sqrt{2M_n}\,\|\tilde{\boldsymbol{\gamma}}_{Gt}\|_2 \leq \sqrt{2M_n}\,\big(\max_{t\in[p^2]} w_t^{-1}\big)\sum_{t=1}^{p^2}w_t\|\tilde{\boldsymbol{\gamma}}_{Gt}\|_2 \\ &\leq \sqrt{2M_n}\,B_{\text{init}}\|\tilde{\boldsymbol{\gamma}}\|_{\bar{\mathcal{G}},2w}\,. \end{aligned} \tag{155}$$

Thus by (153), (154) and (155), if $\bar{\delta} \leq B_{xy}$,

$$A \leq 12B_{\text{init}}^2M_nB_{xy}\,\bar{\delta}\,\|\tilde{\boldsymbol{\gamma}}\|_{\bar{\mathcal{G}},2w}^2\,. \tag{156}$$

By (122) and (123) we have

$$\begin{aligned}\|\tilde{\boldsymbol{\gamma}}\|^2_{\bar{\mathcal{G}},2w} &= \|\tilde{\boldsymbol{\gamma}}_{\mathcal{M}} + \tilde{\boldsymbol{\gamma}}_{\mathcal{M}^\perp}\|^2_{\bar{\mathcal{G}},2w} \\ &= (\|\tilde{\boldsymbol{\gamma}}_{\mathcal{M}}\|_{\bar{\mathcal{G}},2w} + \|\tilde{\boldsymbol{\gamma}}_{\mathcal{M}^\perp}\|_{\bar{\mathcal{G}},2w})^2 \\ &\overset{(122)}{\leq} 16\,\|\tilde{\boldsymbol{\gamma}}_{\mathcal{M}}\|^2_{\bar{\mathcal{G}},2w} \overset{(114)}{\leq} 16\, s_n \|\tilde{\boldsymbol{\gamma}}_{\mathcal{M}}\|_2^2 \leq 16\, s_n \|\tilde{\boldsymbol{\gamma}}\|_2^2\,. \end{aligned} \tag{157}$$

Using (119), (150), (154), (156) and (157), we have

$$\delta\mathcal{L}_r(\tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\theta}}^\diamond) \geq \left(\phi^\diamond_{\min} - 192\, s_n M_n B^2_{init} B_{xy}\bar{\delta}\right)\|\tilde{\boldsymbol{\gamma}}\|_2^2 = \kappa_{\mathcal{L}}\,\|\tilde{\boldsymbol{\gamma}}\|_2^2\,,$$

proving the desired result. ∎

Using [26, Lemma 1] we have Lemma 5.

Lemma 5: Let σ_{xy}, C_0 and N_1 be as in (76), (77) and (78), respectively. Define

$$\mathcal{A} = \max_{k\in[M_n],\, q,\ell\in[p_n]} \left\{\left|[\hat{\boldsymbol{S}}_{xk} - \boldsymbol{S}^\diamond_{xk}]_{q\ell}\right|, \left|[\hat{\boldsymbol{S}}_{yk} - \boldsymbol{S}^\diamond_{yk}]_{q\ell}\right|\right\}.$$

Then for any $\tau > 2$ and sample size $n > N_1$,

$$P\left(\mathcal{A} > C_0\sqrt{\ln(p_n)/K_n}\right) \leq 2/p_n^{\tau-2}\,. \tag{158}$$

Proof: By [26, Lemma 1],

$$P\left(\max_{k,q,\ell}\left|[\hat{\boldsymbol{S}}_{xk} - \boldsymbol{S}^\diamond_{xk}]_{q\ell}\right| > C_{0x}\sqrt{\frac{\ln(p_n)}{K_n}}\right) \leq \frac{1}{p_n^{\tau-2}} \tag{159}$$

$$P\left(\max_{k,q,\ell}\left|[\hat{\boldsymbol{S}}_{yk} - \boldsymbol{S}^\diamond_{yk}]_{q\ell}\right| > C_{0y}\sqrt{\frac{\ln(p_n)}{K_n}}\right) \leq \frac{1}{p_n^{\tau-2}} \tag{160}$$

for any $\tau > 2$ and sample size $n > N_1$ where $C_{0x} = 80\max_{\ell,f}([\boldsymbol{S}^\diamond_x(f)]_{\ell\ell})\sqrt{N_1/\ln(p_n)}$ and $C_{0y} = 80\max_{\ell,f}([\boldsymbol{S}^\diamond_y(f)]_{\ell\ell})\sqrt{N_1/\ln(p_n)}$. Using the union bound,

$$\begin{aligned} &P\left(\mathcal{A} > C_0\sqrt{\ln(p_n)/K_n}\right) \\ &\leq P\left(\max_{k,q,\ell}\left|[\hat{\boldsymbol{S}}_{xk} - \boldsymbol{S}^\diamond_{xk}]_{q\ell}\right| > C_0\sqrt{\ln(p_n)/K_n}\right) \\ &\quad + P\left(\max_{k,q,\ell}\left|[\hat{\boldsymbol{S}}_{yk} - \boldsymbol{S}^\diamond_{yk}]_{q\ell}\right| > C_0\sqrt{\ln(p_n)/K_n}\right) \\ &\leq 2/p_n^{\tau-2} \end{aligned} \tag{161}$$

since $C_0 \geq C_{0x}$ and $C_0 \geq C_{0y}$. ∎

We are now ready to prove Theorem 1.

Proof of Theorem 1: First choose $\bar{\delta}$ to make $\kappa_{\mathcal{L}} > 0$ in Lemma 4. Suppose we take $192 s_n M_n B^2_{init} B_{xy}\bar{\delta} \leq \phi^\diamond_{\min}/4$. Then $\kappa_{\mathcal{L}} \geq 3\phi^\diamond_{\min}/4$. Now pick

$$\bar{\delta} = C_0\sqrt{ln(p_n)/K_n} \;\leq\; \min\left\{B_{xy}, \frac{\phi^\diamond_{\min}}{768 s_n M_n B^2_{init} B_{xy}}\right\}, \tag{162}$$

leading to $192 s_n M_n B^2_{init} B_{xy}\bar{\delta} \leq \phi^\diamond_{\min}/4$. These upper bounds can be ensured by picking appropriate lower bounds to sample size n and invoking Lemma 5. The choice of n specified in (83) satisfies (162) with probability $> 1 - 2/p_n^{\tau-2}$. Using $\bar{\delta} = C_0\sqrt{ln(p_n)/K_n} \leq B_{xy}$, the lower bound on λ_n given in (82) satisfies (123) with $\mathcal{R}^\circledast(\nabla\mathcal{L}_r(\tilde{\boldsymbol{\theta}}^\diamond))$ as in Lemma 3. By [20, Theorem 1], $\hat{\tilde{\boldsymbol{\theta}}}$ given by (120) satisfies

$$\|\hat{\tilde{\boldsymbol{\theta}}} - \tilde{\boldsymbol{\theta}}^\diamond\|_2 \leq \frac{3\lambda_n}{\kappa_{\mathcal{L}}}\Psi(\mathcal{M})\,. \tag{163}$$

The left-side of (163) equals $\|\hat{\tilde{\boldsymbol{\Delta}}} - \tilde{\boldsymbol{\Delta}}^\diamond\|_F$ while the right-side of (163) equals the last term of (84) using $\Psi(\mathcal{M}) \leq \sqrt{s_n}$, $\kappa_{\mathcal{L}} \geq 3\phi^\diamond_{\min}/4$. This proves Theorem 1. ∎

We now turn to the proof of Theorem 2.

Proof of Theorem 2: We have $\|\hat{\tilde{\boldsymbol{\Delta}}}^{(ij)} - (\tilde{\boldsymbol{\Delta}}^\diamond)^{(ij)}\| \leq \|\hat{\tilde{\boldsymbol{\Delta}}} - \tilde{\boldsymbol{\Delta}}^\diamond\|_F \leq \bar{\sigma}_n$ w.h.p. For the edge $\{i,j\} \in \tilde{\mathcal{E}}^\diamond_\Delta$, we have

$$\begin{aligned} \|\hat{\tilde{\boldsymbol{\Delta}}}^{(ij)}\| &= \|(\tilde{\boldsymbol{\Delta}}^\diamond)^{(ij)} + \hat{\tilde{\boldsymbol{\Delta}}}^{(ij)} - (\tilde{\boldsymbol{\Delta}}^\diamond)^{(ij)}\| \\ &\geq \|(\tilde{\boldsymbol{\Delta}}^\diamond)^{(ij)}\| - \|\hat{\tilde{\boldsymbol{\Delta}}}^{(ij)} - (\tilde{\boldsymbol{\Delta}}^\diamond)^{(ij)}\| \\ &\geq \nu - \bar{\sigma}_n \geq 0.6\,\nu \quad \text{for} \quad n \geq N_4 \\ &> \gamma_n\,. \end{aligned} \tag{164}$$

Thus, $\tilde{\mathcal{E}}^\diamond_\Delta \subseteq \hat{\mathcal{E}}_\Delta$. Now consider the set complements $(\tilde{\mathcal{E}}^\diamond_\Delta)^c$ and $\hat{\mathcal{E}}^c_\Delta$. For the edge $\{i.j\} \in (\tilde{\mathcal{E}}^\diamond_\Delta)^c$, $\|(\tilde{\boldsymbol{\Delta}}^\diamond)^{(ij)}\| = 0$. For $n \geq N_4$, w.h.p. we have

$$\begin{aligned} \|\hat{\tilde{\boldsymbol{\Delta}}}^{(ij)}\| &\leq \|(\tilde{\boldsymbol{\Delta}}^\diamond)^{(ij)}\| + \|\hat{\tilde{\boldsymbol{\Delta}}}^{(ij)}\hat{\tilde{\boldsymbol{\Delta}}}^{(ij)} - (\tilde{\boldsymbol{\Delta}}^\diamond)^{(ij)}\| \\ &\leq 0 + \bar{\sigma}_n \leq 0.4\,\nu < \gamma_n\,, \end{aligned} \tag{165}$$

implying that $\{i,j\} \in \hat{\mathcal{E}}^c_\Delta$. Thus, $(\tilde{\mathcal{E}}^\diamond_\Delta)^c \subseteq \hat{\mathcal{E}}^c$, hence $\hat{\mathcal{E}}_\Delta \subseteq \tilde{\mathcal{E}}^\diamond_\Delta$, establishing $\hat{\mathcal{E}}_\Delta = \tilde{\mathcal{E}}^\diamond_\Delta$. ∎

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] R. Dahlhaus, "Graphical interaction models for multivariate time series," *Metrika*, vol. 51, no. 2, pp. 157–172, Aug. 2000.

[5] 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.

[6] 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, 2018.

[7] 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.

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

[9] 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. 2020.

[10] 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.

[11] 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.

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

[13] 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, Barcelona, Spain, Dec. 2016, pp. 595–603.

[14] H. Shu and B. Nan, "Estimation of large covariance and precision matrices from temporally dependent observations," *Ann. Statist.*, vol. 47, no. 3, pp. 1321–1350, Jun. 2019.

[15] J. Songsiri and L. Vandenberghe, "Topology selection in graphical models of autoregressive processes," *J. Mach. Learn. Res.*, vol. 11, pp. 2671–2705, Oct. 2010.

[16] P. J. Schreier and L. L. Scharf, *Statistical Signal Processing of Complex-Valued Data*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[17] 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.

[18] 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.

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

[20] 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.

[21] S. Na, M. Kolar, and O. Koyejo, "Estimating differential latent variable graphical models with applications to brain connectivity," *Biometrika*, vol. 108, pp. 425–442, Jan. 2021.

[22] 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.

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

[24] 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.

[25] A. Jung, G. Hannak, and N. Goertz, "Graphical LASSO based model selection for time series," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1781–1785, Oct. 2015.

[26] J. K. Tugnait, "On sparse high-dimensional graphical model learning for dependent time series," *Signal Process.*, vol. 197, pp. 1–18, Aug. 2022.

[27] J. K. Tugnait, "On conditional independence graph learning from multi-attribute Gaussian dependent time series," *IEEE Open J. Signal Process.*, vol. 6, pp. 705–721, 2025.

[28] J. Krampe and E. Paparoditis, "Frequency domain statistical inference for high-dimensional time series," *J. Amer. Stat. Assoc.*, vol. 120, no. 551, pp. 1580–1592, Jul. 2025.

[29] J. Chang, Q. Jiang, T. McElroy, and X. Shao, "Statistical inference for high-dimensional spectral density matrix," *J. Amer. Stat. Assoc.*, vol. 120, no. 551, pp. 1960–1974, Jul. 2025.

[30] J. K. Tugnait, "Learning multi-attribute differential graphs with non-convex penalties," *IEEE Access*, vol. 13, pp. 67065–67078, 2025.

[31] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[32] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data," *IEEE Signal Process Mag.*, vol. 36, no. 3, pp. 44–63, May 2019.

[33] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.

[34] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," in *Proc. 7th Intern Conf Learn. Represent. (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1–19.

[35] L. Qiao, L. Zhang, S. Chen, and D. Shen, "Data-driven graph construction and graph learning: A review," *Neurocomputing*, vol. 312, pp. 336–351, Oct. 2018.

[36] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 109–127, Apr. 2021.

[37] F. Xia, C. Peng, J. Ren, F. Gozi Febrinanto, R. Luo, V. Saikrishna, S. Yu, and X. Kong, "Graph learning," 2025, *arXiv:2507.05636*.

[38] L.-P. Chen, "Estimation of graphical models: An overview of selected topics," *Int. Stat. Rev.*, vol. 92, no. 2, pp. 194–245, Aug. 2024.

[39] D. Ataee Tarzanagh, B. Hou, Q. Long, L. Shen, and Z. Zhou, "Fairness-aware estimation of graphical models," in *Proc. Adv. Neural Inf. Process. Syst. 37*, Dec. 2024, pp. 17870–17909.

[40] Y. Zhang and Y. Yang, "Joint estimation for multisource Gaussian graphical models based on transfer learning," *Pattern Recognit.*, vol. 158, Feb. 2025, Art. no. 110964.

[41] Z. Kang, C. Peng, Q. Cheng, X. Liu, X. Peng, Z. Xu, and L. Tian, "Structured graph learning for clustering and semi-supervised classification," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107627.

[42] A. Amjad, L.-C. Tai, and H.-T. Chang, "Utilizing enhanced particle swarm optimization for feature selection in gender-emotion detection from English speech signals," *IEEE Access*, vol. 12, pp. 189564–189573, 2024.

[43] A. Amjad, S. Khuntia, H.-T. Chang, and L.-C. Tai, "Multi-domain emotion recognition enhancement: A novel domain adaptation technique for speech-emotion recognition," *IEEE Trans. Audio, Speech Language Process.*, vol. 33, pp. 528–541, 2025.

[44] J. K. Tugnait, "Estimation of differential graphs from time-dependent data," in *Proc. IEEE CAMSAP*, Dec. 2023, pp. 261–265.

[45] K. B. Petersen and M. S. Pedersen. (2012). *The Matrix Cookbook*. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3274

[46] D. R. Brillinger, *Time Series: Data Analysis and Theory*. New York, NY, USA: McGraw-Hill, 1981.

[47] 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.

[48] J. K. Tugnait, "Conditional independence graph estimation from multi-attribute dependent time series," in *Proc. IEEE MLSP*, Sep. 2024, pp. 1–6.

[49] R. Chen, H. Xiao, and D. Yang, "Autoregressive models for matrix-valued time series," *J. Econometrics*, vol. 222, no. 1, pp. 539–560, May 2021.

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 the 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 Urbana–Champaign, 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. He joined the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA, in September 1989, as a Professor, where he is currently the 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.