# Optimization Algorithms for Graph Laplacian Estimation via ADMM and MM

Licheng Zhao, Yiwei Wang, Sandeep Kumar, and Daniel P. Palomar, *Fellow, IEEE*

*Abstract*—In this paper, we study the graph Laplacian estimation problem under a given connectivity topology. We aim at enriching the unified graph learning framework proposed by Egilmez *et al.* and improve the optimality performance of the combinatorial graph Laplacian (CGL) case. We apply the well-known alternating direction method of multipliers (ADMM) and majorization–minimization (MM) algorithmic frameworks and propose two algorithms, namely, GLE-ADMM and GLE-MM, for graph Laplacian estimation. Both algorithms can achieve an optimality gap as low as $10^{-4}$, around three orders of magnitude more accurate than the benchmark. In addition, we find that GLE-ADMM is more computationally efficient in a dense topology (e.g., an almost complete graph), while GLE-MM is more suitable for sparse graphs (e.g., trees). Furthermore, we consider exploiting the leading eigenvectors of the sample covariance matrix as a nominal eigensubspace and propose a third algorithm, named GLENE, which is also based on ADMM. Numerical experiments show that the inclusion of a nominal eigensubspace significantly improves the estimation of the graph Laplacian, which is more evident when the sample size is smaller than or comparable to the problem dimension.

*Index Terms*—Graph learning, Laplacian estimation, nominal eigensubspace, ADMM, Majorization-Minimization.

## I. INTRODUCTION

**G**RAPH signal processing has been a rapidly developing field in recent years, with a wide range of applications such as social, energy, transportation, sensor, and neuronal networks [2]. Its popularity results from the revolutionary way it models data points and their pairwise interconnections. When a collection of data samples are modeled as a graph signal, each sample is treated as a vertex and their pairwise interconnections are represented by a number of edges. Every edge is associated with a weight, and the weight value often reflects the similarity between the connecting vertices. We define a weighted graph as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$, where $\mathcal{V}$ denotes the vertex set with card$(\mathcal{V}) = N$ ($N$ vertices), $\mathcal{E}$ denotes the edge set with card$(\mathcal{E}) = M$ ($M$ edges), and $\mathbf{W} \in \mathbb{R}^{N \times N}$ is the weight matrix. We will focus on a specific type of graph which is undirected and connected (i.e., one connected component only) with no self-loops, so the corresponding weight matrix is symmetric and elementwisely non-negative, with its diagonal elements all being zero. The graph Laplacian, also known as a combinatorial graph Laplacian (see [1, Definition 2]), is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \in \mathbb{R}^{N \times N}, \tag{1}$$

where $\mathbf{D}$ is the degree matrix, which is diagonal in structure with $D_{ii} = \sum_{j=1}^{N} W_{ij}$. The adjacency matrix $\mathbf{A}$ is defined as

$$\mathbf{A} = \mathrm{sgn}(\mathbf{W}) \in \mathbb{R}^{N \times N}, \tag{2}$$

which implies $A_{ij} = 1$ if $W_{ij} > 0$, $A_{ij} = 0$ if $W_{ij} = 0$, and $A_{ii} = 0$.

In most practical scenarios, it is straightforward to derive the vertex set, but the edge set and the associated weight matrix are not readily available. This is either because no reasonable initial graph exists, or only a vague prior is given [3]. Under these circumstances, it is of great significance to learn the graph structure through statistical methods from the available finite data samples. In this paper, we specifically assume the data samples are drawn from a Gaussian Markov Random Field (GMRF) [4]. GMRFs are powerful tools and can be applied to such areas as structural time-series analysis (e.g., autoregressive models), graphical models, semiparametric regression and splines, image analysis, and spatial statistics [4]. The graph structure estimation of a GMRF model naturally amounts to the estimation of the precision matrix (inverse covariance matrix) by means of maximum likelihood estimation. As it is pointed out in the literature, the precision matrix is popularly structured as a graph Laplacian [5], [6] and the corresponding GMRF models are named Laplacian GMRF models. A graph Laplacian is a positive semidefinite (PSD) matrix with non-positive off-diagonal entries and a zero row-sum [7]:

$$\mathcal{L} = \left\{ \mathbf{L} \succeq \mathbf{0} \,\middle|\, \mathbf{L1} = \mathbf{0},\ L_{ij} \le 0,\ i \ne j \right\}, \tag{3}$$

which always corresponds to a graph with non-negative weighted edges [6]. As is mentioned in [6], the significance of the Laplacian GMRF model has been recognized in image reconstruction [8], image segmentation [9], and texture modeling and discrimination [10], [11]. With the aforementioned definitions for $\mathbf{L}$ and $\mathbf{A}$, we can describe the constraint set for graph Laplacians under a given connectivity topology:

$$\mathcal{L}(\mathbf{A}) = \left\{ \mathbf{\Theta} \succeq \mathbf{0} \,\middle|\, \mathbf{\Theta 1} = \mathbf{0},\ \begin{matrix} \Theta_{ij} \le 0 \text{ if } A_{ij} = 1 \\ \Theta_{ij} = 0 \text{ if } A_{ij} = 0 \end{matrix} \text{ for } i \ne j \right\}, \tag{4}$$

which is a subset of $\mathcal{L}$. The graph Laplacian notation is changed to $\mathbf{\Theta}$ so as to align with the majority of the existing works.

### A. Related Works

In the field of GMRF model estimation, the authors of [12] and [13] adopted the $\ell_1$ regularization in pursuit of a sparse graphical model. The estimation problem is to maximize the penalized log-likelihood:

$$\underset{\mathbf{\Theta} \succeq \mathbf{0}}{\text{maximize}} \quad \log \det(\mathbf{\Theta}) - \text{Tr}(\mathbf{S}\mathbf{\Theta}) - \alpha \|\text{vec}(\mathbf{\Theta})\|_1, \quad (5)$$

where $\mathbf{S}$ is the sample covariance matrix. The penalty term $\|\text{vec}(\mathbf{\Theta})\|_1$ promotes elementwise sparsity in $\mathbf{\Theta}$ for the sake of data interpretability and avoiding potential singularity issues [13]. After these two pioneering works, Friedman *et al.* [14] came up with an efficient computational method to solve (5) and proposed the well-known GLasso algorithm, which is a co-ordinate descent procedure by nature.

Up to the time of those early works the Laplacian structure had not yet been imposed on the precision matrix $\mathbf{\Theta}$. When $\mathbf{\Theta}$ has the Laplacian structure, $\det(\mathbf{\Theta})$ equals 0, obtaining minus infinity after the $\log$ operation. To handle this singularity issue, Lake and Tenenbaum [15] lifted the diagonal elements of $\mathbf{\Theta}$ to be $\bar{\mathbf{\Theta}} = \mathbf{\Theta} + \nu \mathbf{I}$. The formulation becomes

$$\underset{\mathbf{\Theta} \succeq \mathbf{0}, \bar{\mathbf{\Theta}}, \nu \geq 0}{\text{minimize}} \quad \text{Tr}(\mathbf{\Theta}\mathbf{S}) - \log \det(\mathbf{\Theta}) + \alpha \|\text{vec}(\mathbf{\Theta})\|_1$$

$$\text{subject to} \quad \mathbf{\Theta} = \bar{\mathbf{\Theta}} + \nu \mathbf{I}$$

$$\bar{\mathbf{\Theta}}\mathbf{1} = \mathbf{0}, \ \bar{\Theta}_{ij} \leq 0, \ i \neq j, \quad (6)$$

and the solution is given as $\mathbf{\Theta}^\star - \nu^\star \mathbf{I}$. Dong *et al.* [7] and Kalofolias [16] also emphasized the Laplacian structure in their graph learning process but modified the maximum penalized log-likelihood formulation as

$$\underset{\mathbf{\Theta} \succeq \mathbf{0}}{\text{maximize}} \text{Tr}(\mathbf{\Theta}\mathbf{S}) + \alpha \|\mathbf{\Theta}\|_F^2$$

$$\text{subject to} \, \mathbf{\Theta}\mathbf{1} = \mathbf{0}, \ \text{Tr}(\mathbf{\Theta}) = N, \ \Theta_{ij} \leq 0, \ i \neq j \quad (7)$$

and

$$\underset{\mathbf{\Theta} \succeq \mathbf{0}}{\text{maximize}} \quad \text{Tr}(\mathbf{\Theta}\mathbf{S}) + \alpha_1 \|\mathbf{\Theta}\|_{F,\text{off}}^2 - \alpha_2 \log \det(\text{Ddiag}(\mathbf{\Theta}))$$

$$\text{subject to} \quad \mathbf{\Theta}\mathbf{1} = \mathbf{0}, \ \Theta_{ij} \leq 0, \ i \neq j. \quad (8)$$

A more reasonable way to estimate a Laplacian structured precision matrix is mentioned in [1]. Egilmez *et al.* [1] proposed a unified framework for Laplacian estimation. They extended the classical graph Laplacian concept into three different classes: Generalized Graph Laplacian (GGL), $\{\mathbf{\Theta} \succeq \mathbf{0} \,|\, \Theta_{ij} \leq 0, \ i \neq j\}$; Diagonally Dominant generalized Graph Laplacian (DDGL), $\{\mathbf{\Theta} \succeq \mathbf{0} \,|\, \mathbf{\Theta}\mathbf{1} \geq \mathbf{0}, \ \Theta_{ij} \leq 0, \ i \neq j\}$; and Combinatorial Graph Laplacian (CGL), the same as the graph Laplacian matrix in (3). A total of two algorithms were proposed, one for GGL and DDGL and the other for CGL. We find that the one for GGL and DDGL is efficient and gives empirically satisfactory numerical performance, but the other for CGL, is not so accurate in terms of optimality on most occasions and may violate the constraint set from time to time. This results from the heuristic operations mentioned in [1, Algorithm 2, Line 13 to 17]. Interested readers may refer to [17] for extensions of this unified framework.

### B. Contribution

The major contributions of this paper are as follows:
1) We propose two algorithms for graph Laplacian estimation under a given connectivity topology, namely GLE-ADMM and GLE-MM. Both algorithms can achieve an optimality gap as low as $10^{-4}$, around three orders of magnitude more accurate than the benchmark CGL. Moreover, we find that GLE-ADMM is more computationally efficient in a dense topology (e.g., an almost complete graph), while GLE-MM is more suitable for sparse graphs (e.g., trees).
2) We additionally consider exploiting the leading eigenvectors of the sample covariance matrix as a nominal eigensubspace. This improves the estimation of the graph Laplacian when the sample size is smaller than or comparable to the problem dimension, as is suggested by the simulation results in Section VI-B1. We propose an algorithm named GLENE based on the Alternating Direction Method of Multipliers (ADMM) for the inclusion of a nominal eigensubspace. The optimality gap with respect to the CVX toolbox is less than $10^{-4}$. In a real-data experiment, we show that GLENE is able to reveal the strong correlations between stocks, while achieving a high sparsity ratio.

### C. Organization and Notation

The rest of the paper is organized as follows. In Section II, we present the problem formulation of graph Laplacian estimation. In Section III, we introduce an algorithmic solution for graph Laplacian estimation based on the ADMM framework. In Section IV, we revisit the graph Laplacian estimation problem and propose an alternative solution via the Majorization-Minimization (MM) framework. In Section V, we study the graph Laplacian estimation problem with the inclusion of a nominal eigensubspace. Section VI presents numerical results, and the conclusions are drawn in Section VII.

The following notation is adopted. Boldface upper-case letters represent matrices, boldface lower-case letters denote column vectors, and standard lower-case or upper-case letters stand for scalars. $\mathbb{R}$ denotes the real field. $\odot$ stands for the Hadamard product. $\text{sgn}(x) = x/|x|, \text{sgn}(0) = 0$, $[x]_+ = \max(x, 0)$, $[x]_- = \min(x, 0)$, $[\mathbf{X}]_+ = \max(\mathbf{X}, \mathbf{0})$, and $[\mathbf{X}]_- = \min(\mathbf{X}, \mathbf{0})$. $\mathbf{X} \geq \mathbf{0}$ means $\mathbf{X}$ is elementwisely larger than 0. $[\mathbf{X}]_{\text{PSD}} = \mathbf{U}[\mathbf{\Lambda}]_+ \mathbf{U}^T$, with $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ being the eigenvalue decomposition of $\mathbf{X}$. $\|\cdot\|_p$ denotes the $\ell_p$-norm of a vector. $\nabla(\cdot)$ represents the gradient of a multivariate function. $\mathbf{1}$ stands for the all-one vector, and $\mathbf{I}$ stands for the identity matrix. $\mathbf{X}^T$, $\mathbf{X}^{-1}$, $\mathbf{X}^\dagger$, $\text{Tr}(\mathbf{X})$, and $\det(\mathbf{X})$ denote the transpose, inverse, pseudo-inverse, trace, and determinant of $\mathbf{X}$, respectively. $\mathbf{X} \succeq \mathbf{0}$ means $\mathbf{X}$ is positive semidefinite. $\text{diag}(\mathbf{X})$ is the vector consisting of all the diagonal elements of matrix $\mathbf{X}$. $\text{Diag}(\mathbf{x})$ is a diagonal matrix with $\mathbf{x}$ filling its principal diagonal. $\text{Ddiag}(\mathbf{X})$ is a diagonal matrix with the diagonal elements of $\mathbf{X}$ filling its principal diagonal. $\|\mathbf{X}\|_F$ is the Frobenius norm of $\mathbf{X}$, and $\|\mathbf{X}\|_{F,\text{off}}$ is the Frobenius norm of $\mathbf{X} - \text{Ddiag}(\mathbf{X})$. The cardinality of the set $\mathcal{X}$ is represented as $\text{card}(\mathcal{X})$. The superscript $\star$ represents the optimal solution of some optimization problem. Whenever arithmetic operators

$(\sqrt{\cdot}, \cdot/\cdot, \cdot^2, \cdot^{-1},$ etc.) are applied to vectors or matrices, they are elementwise operations.

## II. PROBLEM STATEMENT

Suppose we obtain a number of samples $\{\mathbf{x}_i\}_{i=1}^T$ from a GMRF model. We are able to compute a certain data statistic $\mathbf{S} \in \mathbb{R}^{N \times N}$ (e.g., sample covariance matrix) thereafter. Our goal is to estimate the graph structure of the model, so we carry out the graph learning process, which typically consists of two steps: topology inference and weight estimation [18]. In this paper, we assume the graph topology is given, i.e., the adjacency matrix $\mathbf{A}$ is known, and we focus on weight estimation. One of the most extensively studied problems in weight estimation is to estimate the graph Laplacian. A seemingly plausible problem formulation for graph Laplacian estimation is given as follows:

$$\underset{\mathbf{\Theta}}{\text{minimize}} \quad \text{Tr}(\mathbf{\Theta S}) - \log \det(\mathbf{\Theta}) + \alpha \|\text{vec}(\mathbf{\Theta})\|_1$$

$$\text{subject to} \quad \mathbf{\Theta} \in \mathcal{L}(\mathbf{A}), \tag{9}$$

where $\alpha > 0$ is the regularization parameter. Now that $\mathbf{\Theta}$ satisfies the Laplacian constraints, the off-diagonal elements of $\mathbf{\Theta}$ are non-positive and the diagonal elements are non-negative, so

$$\|\text{vec}(\mathbf{\Theta})\|_1 = \text{Tr}(\mathbf{\Theta H}), \tag{10}$$

where $\mathbf{H} = 2\mathbf{I} - \mathbf{1}\mathbf{1}^T$. Thus, the objective function becomes

$$\begin{aligned} &\text{Tr}(\mathbf{\Theta S}) - \log \det(\mathbf{\Theta}) + \alpha \|\text{vec}(\mathbf{\Theta})\|_1 \\ &= \text{Tr}(\mathbf{\Theta S}) - \log \det(\mathbf{\Theta}) + \alpha \text{Tr}(\mathbf{\Theta H}) \\ &= \text{Tr}(\mathbf{\Theta}(\mathbf{S} + \alpha \mathbf{H})) - \log \det(\mathbf{\Theta}) \\ &\triangleq \text{Tr}(\mathbf{\Theta K}) - \log \det(\mathbf{\Theta}), \end{aligned} \tag{11}$$

where $\mathbf{K} \triangleq \mathbf{S} + \alpha \mathbf{H}$. However, once the Laplacian constraints are satisfied, $\mathbf{\Theta}$ is not positive definite because $\mathbf{1}^T \mathbf{\Theta} \mathbf{1} = 0$, which leads to $\log \det(\mathbf{\Theta})$ being unbounded below. To address the singularity issue, Egilmez *et al.* [1] proposed to modify $\log \det(\mathbf{\Theta})$ as $\log \det(\mathbf{\Theta} + \mathbf{J})$, where[1] $\mathbf{J} = \frac{1}{N}\mathbf{1}\mathbf{1}^T$, and the reformulated problem takes the following form:

$$\underset{\mathbf{\Theta}}{\text{minimize}} \quad \text{Tr}(\mathbf{\Theta K}) - \log \det(\mathbf{\Theta} + \mathbf{J})$$

$$\text{subject to} \quad \mathbf{\Theta} \in \mathcal{L}(\mathbf{A}). \tag{12}$$

Its validity holds if the graph topology has only one connected component [19]. This problem is solved with [1, Algorithm 2], otherwise called CGL, in the existing literature, but the optimality performance of this algorithm is not satisfactory, so we aim at improving the CGL algorithm.

## III. GRAPH LAPLACIAN ESTIMATION: AN ADMM APPROACH

First we study the constraint set $\mathcal{L}(\mathbf{A})$, which is written as follows:

$$\begin{cases} \mathbf{\Theta} \succeq \mathbf{0}, \ \mathbf{\Theta}\mathbf{1} = \mathbf{0} \\ \Theta_{ij} \leq 0 \ \text{if} \ A_{ij} = 1 \\ \Theta_{ij} = 0 \ \text{if} \ A_{ij} = 0 \end{cases} \text{for } i \neq j. \tag{13}$$

We further suppose the graph has no self loops, so the diagonal elements of $\mathbf{A}$ are all zero. Then, the constraint set $\mathcal{L}(\mathbf{A})$ can be compactly rewritten in the following way:

$$\begin{cases} \mathbf{\Theta} \succeq \mathbf{0}, \ \mathbf{\Theta}\mathbf{1} = \mathbf{0} \\ \mathbf{\Theta} - \mathbf{C} = \mathbf{0} \\ \left.\begin{array}{l} \mathbf{I} \odot \mathbf{C} \geq \mathbf{0} \\ \mathbf{B} \odot \mathbf{C} = \mathbf{0} \\ \mathbf{A} \odot \mathbf{C} \leq \mathbf{0} \end{array}\right\} \mathbf{C} \in \mathcal{C}, \end{cases} \tag{14}$$

where

$$\mathbf{B} = \mathbf{1}\mathbf{1}^T - \mathbf{I} - \mathbf{A}. \tag{15}$$

The constraint $\mathbf{I} \odot \mathbf{C} \geq \mathbf{0}$ is implied from the constraint $\mathbf{\Theta} \succeq \mathbf{0}$.

Next we will present an equivalent form of the constraints $\mathbf{\Theta}\mathbf{1} = \mathbf{0}$ and $\mathbf{\Theta} \succeq \mathbf{0}$:

$$\begin{aligned} &\mathbf{\Theta} \succeq \mathbf{0}, \ \mathbf{\Theta}\mathbf{1} = \mathbf{0} \\ \iff &\mathbf{\Theta} = \mathbf{P}\mathbf{\Xi}\mathbf{P}^T, \ \mathbf{\Xi} \succeq \mathbf{0}, \end{aligned} \tag{16}$$

where $\mathbf{P} \in \mathbb{R}^{N \times (N-1)}$ is the orthogonal complement of $\mathbf{1}$, i.e., $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ and $\mathbf{P}^T\mathbf{1} = \mathbf{0}$. Note that the choice of $\mathbf{P}$ is non-unique; if $\mathbf{P}_0$ satisfies the aforementioned two conditions, $\mathbf{P}_0\mathbf{U}$ will also do if $\mathbf{U} \in \mathbb{R}^{(N-1) \times (N-1)}$ is unitary. With the equivalent form of $\mathbf{\Theta}$, we can rewrite the objective function as follows:

$$\text{Tr}(\mathbf{\Theta K}) = \text{Tr}\left(\mathbf{\Xi}\tilde{\mathbf{K}}\right), \tag{17}$$

where $\tilde{\mathbf{K}} = \mathbf{P}^T\mathbf{K}\mathbf{P}$. We also have

$$\begin{aligned} &\log \det(\mathbf{\Theta} + \mathbf{J}) \\ &= \log \det\left(\mathbf{P}\mathbf{\Xi}\mathbf{P}^T + \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) \\ &= \log \det\left(\left[\mathbf{P}, \mathbf{1}/\sqrt{N}\right]\begin{bmatrix}\mathbf{\Xi} & \\ & 1\end{bmatrix}\begin{bmatrix}\mathbf{P}^T \\ \mathbf{1}^T/\sqrt{N}\end{bmatrix}\right) \\ &= \log \det(\mathbf{\Xi}). \end{aligned} \tag{18}$$

Thus, the problem formulation changes to

$$\underset{\mathbf{\Xi}, \mathbf{C}}{\text{minimize}} \quad \text{Tr}\left(\mathbf{\Xi}\tilde{\mathbf{K}}\right) - \log \det(\mathbf{\Xi})$$

$$\text{subject to} \quad \mathbf{\Xi} \succeq \mathbf{0}$$

$$\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C} = \mathbf{0}$$

$$\left.\begin{array}{l} \mathbf{I} \odot \mathbf{C} \geq \mathbf{0} \\ \mathbf{B} \odot \mathbf{C} = \mathbf{0} \\ \mathbf{A} \odot \mathbf{C} \leq \mathbf{0} \end{array}\right\} \mathbf{C} \in \mathcal{C}. \tag{19}$$

We will solve (19) with the ADMM algorithmic framework.

### A. The ADMM Framework

The ADMM algorithm is aimed at solving problems in the following format:

$$\underset{\mathbf{x}, \mathbf{z}}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{z})$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \tag{20}$$

with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times m}$, and $\mathbf{c} \in \mathbb{R}^p$. $f$ and $g$ are convex functions. The augmented Lagrangian of (20)

---

[1]This modification is justified in [1, Prop. 1].

is given as

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c})$$
$$+ (\rho/2)\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2. \quad (21)$$

The ADMM framework is summarized as follows:

---

**Require:** $l = 0$, $\mathbf{y}^{(0)}$, and $\mathbf{z}^{(0)}$.
1: **repeat**
2:    $\mathbf{x}^{(l+1)} = \arg\min_{\mathbf{x}\in\mathcal{X}} L_\rho(\mathbf{x}, \mathbf{z}^{(l)}, \mathbf{y}^{(l)})$;
3:    $\mathbf{z}^{(l+1)} = \arg\min_{\mathbf{z}\in\mathcal{Z}} L_\rho(\mathbf{x}^{(l+1)}, \mathbf{z}, \mathbf{y}^{(l)})$;
4:    $\mathbf{y}^{(l+1)} = \mathbf{y}^{(l)} + \rho(\mathbf{A}\mathbf{x}^{(l+1)} + \mathbf{B}\mathbf{z}^{(l+1)} - \mathbf{c})$;
5:    $l = l + 1$;
6: **until** convergence

---

The convergence of ADMM is obtained if the following conditions are satisfied:
1) **epi** $f = \{(\mathbf{x}, t) \in \mathbb{R}^n \times \mathbb{R} \mid f(\mathbf{x}) \le t\}$ and **epi** $g = \{(\mathbf{z}, s) \in \mathbb{R}^n \times \mathbb{R} \mid g(\mathbf{z}) \le s\}$ are both closed nonempty convex sets;
2) The unaugmented Lagrangian $L_0$ has a saddle point.
Detailed convergence results are elaborated in [20, Sec. 3.2].

### B. Implementation of ADMM

We derive the (partial) augmented Lagrangian:

$$\mathcal{L}(\mathbf{\Xi}, \mathbf{C}, \mathbf{Y}) = \text{Tr}\left(\mathbf{\Xi}\tilde{\mathbf{K}}\right) - \log\det(\mathbf{\Xi})$$
$$+ \text{Tr}\left(\mathbf{Y}^T(\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C})\right) + \frac{\rho}{2}\|\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C}\|_F^2. \quad (22)$$

We treat $\mathbf{\Xi}$ and $\mathbf{C}$ as primal variables and define $\mathbf{Y}$ as the dual variable with respect to the constraint $\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C} = \mathbf{0}$. The constraints $\mathbf{\Xi} \succeq \mathbf{0}$ and $\mathbf{C} \in \mathcal{C}$ are not relaxed, so they do not show up in the augmented Lagrangian. The first two steps in the ADMM algorithm are to find the minimizer of the augmented Lagrangian with respect to $\mathbf{\Xi}$ and $\mathbf{C}$, respectively, with the other primal and dual variables fixed, i.e., (for simple notation, the update variable has a superscript "+")

$$\begin{cases} \mathbf{\Xi}^+ = \arg\min_{\mathbf{\Xi}\succeq\mathbf{0}} \mathcal{L}(\mathbf{\Xi}, \mathbf{C}, \mathbf{Y}) \\ \mathbf{C}^+ = \arg\min_{\mathbf{C}\in\mathcal{C}} \mathcal{L}(\mathbf{\Xi}^+, \mathbf{C}, \mathbf{Y}). \end{cases} \quad (23)$$

*1) Update of $\mathbf{\Xi}$:*

$$\mathbf{\Xi}^+ = \arg\min_{\mathbf{\Xi}\succeq\mathbf{0}} \mathcal{L}(\mathbf{\Xi}, \mathbf{C}, \mathbf{Y})$$
$$= \arg\min_{\mathbf{\Xi}\succeq\mathbf{0}} \text{Tr}\left(\mathbf{\Xi}\tilde{\mathbf{K}}\right) - \log\det(\mathbf{\Xi})$$
$$+ \text{Tr}\left(\mathbf{P}^T\mathbf{Y}^T\mathbf{P}\mathbf{\Xi}\right) + \frac{\rho}{2}\|\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C}\|_F^2$$
$$= \arg\min_{\mathbf{\Xi}\succeq\mathbf{0}} \frac{\rho}{2}\left\|\mathbf{\Xi} + \frac{1}{\rho}\left(\tilde{\mathbf{K}} + \tilde{\mathbf{Y}} - \rho\tilde{\mathbf{C}}\right)\right\|_F^2$$
$$- \log\det(\mathbf{\Xi}), \quad (24)$$

with $\tilde{\mathbf{Y}} = \mathbf{P}^T\mathbf{Y}\mathbf{P}$ and $\tilde{\mathbf{C}} = \mathbf{P}^T\mathbf{C}\mathbf{P}$. The next step is to compute the minimizer to a problem of this format: $\frac{\rho}{2}\|\mathbf{\Theta} + \mathbf{X}\|_F^2 - \log\det(\mathbf{\Theta})$, where the variable is $\mathbf{\Theta}$. Thus, we introduce the following supporting lemma.

---

**Algorithm 1:** ADMM-Based Graph Laplacian Estimation (GLE-ADMM).

**Require:** Initialization: $\mathbf{K}$, symmetric $\mathbf{Y}^{(0)}$ and $\mathbf{C}^{(0)}$, $\rho > 0$, $l = 0$
1: **repeat**
2:    Eigenvalue decomposition: $\frac{1}{\rho}\mathbf{P}^T(\mathbf{K} + \mathbf{Y}^{(l)} - \rho\mathbf{C}^{(l)})\mathbf{P} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$;
3:    $\mathbf{D}$ is diagonal, with $D_{ii} = \frac{-\rho\Lambda_{ii}+\sqrt{\rho^2\Lambda_{ii}^2+4\rho}}{2\rho}$;
4:    $\mathbf{\Xi}^{(l+1)} = \mathbf{U}\mathbf{D}\mathbf{U}^T$;
5:    $\mathbf{\Theta}^{(l+1)} = \mathbf{P}\mathbf{\Xi}^{(l+1)}\mathbf{P}^T$;
6:    $\mathbf{C}^{(l+1)} = \mathbf{I} \odot [\frac{1}{\rho}\mathbf{Y}^{(l)} + \mathbf{\Theta}^{(l+1)}]_+ + \mathbf{A} \odot [\frac{1}{\rho}\mathbf{Y}^{(l)} + \mathbf{\Theta}^{(l+1)}]_-$;
7:    $\mathbf{Y}^{(l+1)} = \mathbf{Y}^{(l)} + \rho(\mathbf{\Theta}^{(l+1)} - \mathbf{C}^{(l+1)})$;
8:    $l = l + 1$;
9: **until** convergence

---

*Lemma 1 ([20, Chap. 6.5]):* The solution to $\min_{\mathbf{\Theta}\succ\mathbf{0}} \frac{\rho}{2}\|\mathbf{\Theta} + \mathbf{X}\|_F^2 - \log\det(\mathbf{\Theta})$ is $\mathbf{\Theta}^\star = \mathbf{U}\mathbf{D}\mathbf{U}^T$, where $\mathbf{U}$ comes from the eigenvalue decomposition of $\mathbf{X}$, i.e., $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, and $\mathbf{D}$ is a diagonal matrix with

$$D_{ii} = \frac{-\rho\Lambda_{ii} + \sqrt{\rho^2\Lambda_{ii}^2 + 4\rho}}{2\rho}. \quad (25)$$

Applying Lemma 1, we can obtain

$$\mathbf{\Xi}^+ = \mathbf{U}\mathbf{D}\mathbf{U}^T, \quad (26)$$

where $\mathbf{U}$ comes from the eigenvalue decomposition of $\frac{1}{\rho}(\tilde{\mathbf{K}} + \tilde{\mathbf{Y}} - \rho\tilde{\mathbf{C}}) = \frac{1}{\rho}\mathbf{P}^T(\mathbf{K} + \mathbf{Y} - \rho\mathbf{C})\mathbf{P}$, i.e., $\frac{1}{\rho}\mathbf{P}^T(\mathbf{K} + \mathbf{Y} - \rho\mathbf{C})\mathbf{P} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, and $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \frac{-\rho\Lambda_{ii}+\sqrt{\rho^2\Lambda_{ii}^2+4\rho}}{2\rho}$.

*2) Update of $\mathbf{C}$:*

$$\mathbf{C}^+ = \arg\min_{\mathbf{C}\in\mathcal{C}} \mathcal{L}(\mathbf{\Xi}^+, \mathbf{C}, \mathbf{Y})$$
$$= \arg\min_{\mathbf{C}\in\mathcal{C}} -\text{Tr}(\mathbf{Y}^T\mathbf{C}) + \frac{\rho}{2}\|\mathbf{\Theta}^+ - \mathbf{C}\|_F^2, \quad (27)$$

where $\mathbf{\Theta}^+ = \mathbf{P}\mathbf{\Xi}^+\mathbf{P}^T$. Now we need another supporting lemma to find the minimizer.

*Lemma 2:* The solution to $\min_{\mathbf{C}\in\mathcal{C}} -\text{Tr}(\mathbf{Y}^T\mathbf{C}) + \frac{\rho}{2}\|\mathbf{X} - \mathbf{C}\|_F^2$ is $\mathbf{C}^\star = \mathbf{I} \odot [\frac{1}{\rho}\mathbf{Y} + \mathbf{X}]_+ + \mathbf{A} \odot [\frac{1}{\rho}\mathbf{Y} + \mathbf{X}]_-$, where $\mathcal{C} = \{\mathbf{C}|\mathbf{I} \odot \mathbf{C} \ge \mathbf{0},\ \mathbf{B} \odot \mathbf{C} = \mathbf{0},\ \mathbf{A} \odot \mathbf{C} \le \mathbf{0}\}$.

*Proof:* See Appendix A. ∎

Applying Lemma 2, we can obtain the update of $\mathbf{C}$:

$$\mathbf{C}^+ = \mathbf{I} \odot \left[\frac{1}{\rho}\mathbf{Y} + \mathbf{\Theta}^+\right]_+ + \mathbf{A} \odot \left[\frac{1}{\rho}\mathbf{Y} + \mathbf{\Theta}^+\right]_-. \quad (28)$$

The last step of the ADMM algorithm is the dual update, which is as simple as

$$\mathbf{Y}^+ = \mathbf{Y} + \rho\left(\mathbf{\Theta}^+ - \mathbf{C}^+\right), \quad (29)$$

with $\mathbf{\Theta}^+ = \mathbf{P}\mathbf{\Xi}^+\mathbf{P}^T$. We summarize the ADMM solution in Algorithm 1.

*Remark 1 (Implementation Tips):* When we implement the ADMM framework, the choice of the parameter $\rho$ is often a

involved task. In [20, Sec. 3.4.1], the authors suggest an adaptive update scheme for $\rho$ so that it varies in every iteration and becomes less dependent on the initial choice. The update rule is [20, eq. (3.13)]: given $\rho^{(0)}$,

$$\rho^{(l+1)} = \begin{cases} \tau^{\text{inc}} \rho^{(l)} & \left\| \mathbf{r}^{(l)} \right\|_2 > \mu \left\| \mathbf{s}^{(l)} \right\|_2 \\ \rho^{(l)} / \tau^{\text{dec}} & \left\| \mathbf{s}^{(l)} \right\|_2 > \mu \left\| \mathbf{r}^{(l)} \right\|_2 \\ \rho^{(l)} & \text{otherwise,} \end{cases} \quad (30)$$

where $\mu > 1$, $\tau^{\text{inc}} > 1$, and $\tau^{\text{dec}} > 1$ are tuning parameters and $\mathbf{r}^{(l)} = \mathbf{A}\mathbf{x}^{(l)} + \mathbf{B}\mathbf{z}^{(l)} - \mathbf{c}$ and $\mathbf{s}^{(l)} = \rho\mathbf{A}^T\mathbf{B}(\mathbf{z}^{(l)} - \mathbf{z}^{(l-1)})$ (following the notation in Section III-A). We strongly recommend this simple scheme because it indeed accelerates the empirical convergence speed in our implementation.

### C. Computational Complexity

We present an analysis on the computational complexity of Algorithm 1 in this subsection. The analysis is carried out on a per-iteration basis. In every iteration, we update $\boldsymbol{\Xi}$, $\mathbf{C}$, and $\mathbf{Y}$. The update of $\boldsymbol{\Xi}$ involves the following costly steps: i) matrix multiplication $\frac{1}{\rho}\mathbf{P}^T(\mathbf{K} + \mathbf{Y}^{(l)} - \rho\mathbf{C}^{(l)})\mathbf{P}$: $\mathcal{O}(N^3)$, ii) eigenvalue decomposition: $\mathcal{O}(N^3)$, and iii) matrix multiplication $\mathbf{U}\mathbf{D}\mathbf{U}^T$: $\mathcal{O}(N^3)$. The costly step in updating $\mathbf{C}$ is merely the matrix multiplication $\mathbf{P}\boldsymbol{\Xi}^{(l+1)}\mathbf{P}^T$: $\mathcal{O}(N^3)$ since the Hadamard product and the arithmetic operations $[\cdot]_+$ and $[\cdot]_-$ only take $\mathcal{O}(N^2)$. The update of $\mathbf{Y}$ costs $\mathcal{O}(N^2)$. Therefore, the per-iteration complexity of Algorithm 1 is $\mathcal{O}(N^3)$, resulting from six matrix multiplications and one eigenvalue decomposition.

### IV. GRAPH LAPLACIAN ESTIMATION REVISITED: AN MM ALTERNATIVE

We have just solved the graph Laplacian estimation problem with an ADMM approach. The ADMM solution is more suitable for a dense topology, i.e., all the samples (nodes) are connected as an almost complete graph. If the number of non-zero off-diagonal elements in the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ reaches $\mathcal{O}(N^2)$ (or, equivalently, the edge number $M$ reaches $\mathcal{O}(N^2)$), we can unhesitantly resort to the ADMM approach. However, when the graph is sparse, i.e., $M = \mathcal{O}(N)$, the ADMM solution may give way to a more efficient method. We start from the following toy example to gain some insight.

*Example 3:* Suppose we have a $3 \times 3$ Laplacian matrix (for sanity check, please refer to eq. (4)):

$$\begin{bmatrix} 3 & -1 & -2 \\ -1 & 4 & -3 \\ -2 & -3 & 5 \end{bmatrix}.$$

We can perform a special rank-one decomposition to this matrix (different from the traditional eigenvalue decomposition); see eq. (31) shown at the bottom of this page.

For a general graph where there are $M$ edges and the $m$th edge connects vertex $i_m$ and $j_m$, we can always perform the same decomposition on its graph Laplacian $\boldsymbol{\Theta}$:

$$\begin{aligned} \boldsymbol{\Theta} &= \sum_{m=1}^{M} W_{i_m j_m} \left( \mathbf{e}_{i_m} \mathbf{e}_{i_m}^T + \mathbf{e}_{j_m} \mathbf{e}_{j_m}^T - \mathbf{e}_{i_m} \mathbf{e}_{j_m}^T - \mathbf{e}_{j_m} \mathbf{e}_{i_m}^T \right) \\ &= \sum_{m=1}^{M} W_{i_m j_m} \left( \mathbf{e}_{i_m} - \mathbf{e}_{j_m} \right) \left( \mathbf{e}_{i_m} - \mathbf{e}_{j_m} \right)^T \\ &\triangleq \sum_{m=1}^{M} W_{i_m j_m} \boldsymbol{\xi}_{i_m j_m} \boldsymbol{\xi}_{i_m j_m}^T \\ &\triangleq \mathbf{E}\text{Diag}\left( \mathbf{w} \right) \mathbf{E}^T, \end{aligned} \quad (32)$$

where $\mathbf{w} = \{W_{i_m j_m}\}_{m=1}^M$ represents the weights on the edges. The matrix $\mathbf{E}$, known as the incidence matrix, can be inferred from the adjacency matrix $\mathbf{A}$. This decomposition technique was mentioned in [18] as well. The advantage of this decomposition is the simplification of the Laplacian constraints; they are naturally satisfied if and only if $\mathbf{w} \geq \mathbf{0}$. One drawback of this decomposition is that, when the length of $\mathbf{w}$ reaches $\mathcal{O}(N^2)$, the computational cost will be prohibitively high. Given $\mathbf{E}$ and $\mathbf{w}$, a simple computation of $\boldsymbol{\Theta}$ takes up to $\mathcal{O}(N^5)$ operations. To this point, we can see that the efficiency of this decomposition technique depends heavily on the sparsity level of the Laplacian matrix.

When we adopt this decomposition, the original problem formulation (12) becomes

$$\begin{aligned} \underset{\mathbf{w} \geq \mathbf{0}}{\text{minimize}} \quad &\text{Tr}\left( \mathbf{E}\text{Diag}\left( \mathbf{w} \right) \mathbf{E}^T \mathbf{K} \right) \\ &- \log\det\left( \mathbf{E}\text{Diag}\left( \mathbf{w} \right) \mathbf{E}^T + \mathbf{J} \right), \end{aligned} \quad (33)$$

with $\mathbf{J} = \frac{1}{N}\mathbf{1}\mathbf{1}^T$. It is obvious that $\mathbf{E}\text{Diag}(\mathbf{w})\mathbf{E}^T + \frac{1}{N}\mathbf{1}\mathbf{1}^T = [\mathbf{E}, \mathbf{1}]\text{Diag}([\mathbf{w}^T, 1/N]^T)[\mathbf{E}, \mathbf{1}]^T \triangleq \mathbf{G}\text{Diag}([\mathbf{w}^T, 1/N]^T)\mathbf{G}^T$,

$$\begin{aligned} \begin{bmatrix} 3 & -1 & -2 \\ -1 & 4 & -3 \\ -2 & -3 & 5 \end{bmatrix} &= 1 \times \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 2 \times \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} + 3 \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \\ &= 1 \times \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} + 2 \times \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} + 3 \times \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 2 & \\ & & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix}^T. \end{aligned} \quad (31)$$

so the objective can be simplified as $\mathrm{Tr}(\mathbf{E}\mathrm{Diag}(\mathbf{w})\mathbf{E}^T\mathbf{K}) - \log\det(\mathbf{G}\mathrm{Diag}([\mathbf{w}^T, 1/N]^T)\mathbf{G}^T)$. We will apply the MM algorithmic framework to solve (33).

### A. The MM Framework

The MM method can be applied to solve the following general optimization problem [21]–[25]:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{X}, \end{array} \qquad (34)$$

where $f$ is differentiable. Instead of minimizing $f(\mathbf{x})$ directly, we consider successively solving a series of simple optimization problems. The algorithm initializes at some feasible starting point $\mathbf{x}^{(0)}$, and then iterates as $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ until some convergence criterion is met. For any iteration, say, the $l$th iteration, the update rule is

$$\mathbf{x}^{(l+1)} \in \arg \min_{\mathbf{x}\in\mathcal{X}} \bar{f}\left(\mathbf{x}; \mathbf{x}^{(l)}\right), \qquad (35)$$

where $\bar{f}(\mathbf{x}; \mathbf{x}^{(l)})$ (assumed to be smooth) is the majorizing function of $f(\mathbf{x})$ at $\mathbf{x}^{(l)}$. $\bar{f}(\mathbf{x}; \mathbf{x}^{(l)})$ must satisfy the following conditions so as to claim convergence [26]:

A1) $\bar{f}(\mathbf{x}; \mathbf{x}^{(l)}) \geq f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$;
A2) $\bar{f}(\mathbf{x}^{(l)}; \mathbf{x}^{(l)}) = f(\mathbf{x}^{(l)})$;
A3) $\nabla\bar{f}(\mathbf{x}^{(l)}; \mathbf{x}^{(l)}) = \nabla f(\mathbf{x}^{(l)})$ and
A4) $\bar{f}(\mathbf{x}; \mathbf{x}^{(l)})$ is continuous in both $\mathbf{x}$ and $\mathbf{x}^{(l)}$.

One acceleration scheme of the MM framework, known as SQUAREM, can be found in [27] and [28].

### B. Implementation of MM

The fundamental part of the MM method is the construction of a majorizing function. The involved part lies in the majorization of $-\log\det(\mathbf{G}\mathrm{Diag}([\mathbf{w}^T, 1/N]^T)\mathbf{G}^T)$. We start from the following basic inequality:

$$\log\det(\mathbf{X}) \leq \log\det(\mathbf{X}_0) + \mathrm{Tr}\left(\mathbf{X}_0^{-1}(\mathbf{X} - \mathbf{X}_0)\right), \qquad (36)$$

which is due to the concavity of the log-determinant function [29]. Thus,

$$\begin{aligned} &-\log\det\left(\mathbf{G}\mathbf{X}\mathbf{G}^T\right) = \log\det\left(\left(\mathbf{G}\mathbf{X}\mathbf{G}^T\right)^{-1}\right) \\ &\leq \log\det\left(\left(\mathbf{G}\mathbf{X}_0\mathbf{G}^T\right)^{-1}\right) + \mathrm{Tr}\Bigg(\left[\left(\mathbf{G}\mathbf{X}_0\mathbf{G}^T\right)^{-1}\right]^{-1} \cdot \\ &\quad \left(\left(\mathbf{G}\mathbf{X}\mathbf{G}^T\right)^{-1} - \left(\mathbf{G}\mathbf{X}_0\mathbf{G}^T\right)^{-1}\right)\Bigg) \\ &= \mathrm{Tr}\left(\mathbf{F}_0\left(\mathbf{G}\mathbf{X}\mathbf{G}^T\right)^{-1}\right) + \text{const.}, \end{aligned} \qquad (37)$$

where $\mathbf{F}_0 = \mathbf{G}\mathbf{X}_0\mathbf{G}^T$. We substitute $\mathrm{Diag}([\mathbf{w}^T, 1/N]^T)$ for $\mathbf{X}$, and the minimization problem becomes

$$\begin{aligned} \underset{\mathbf{w}\geq\mathbf{0}}{\text{minimize}} \quad &\mathrm{Tr}\left(\mathbf{E}\mathrm{Diag}(\mathbf{w})\mathbf{E}^T\mathbf{K}\right) \\ &+ \mathrm{Tr}\left(\mathbf{F}_0\left(\mathbf{G}\mathrm{Diag}\left([\mathbf{w}^T, 1/N]^T\right)\mathbf{G}^T\right)^{-1}\right), \end{aligned} \qquad (38)$$

with $\mathbf{F}_0 = \mathbf{G}\mathbf{X}_0\mathbf{G}^T = \mathbf{G}\mathrm{Diag}([\mathbf{w}_0^T, 1/N]^T)\mathbf{G}^T$. This minimization problem does not yield a simple closed-form solution

---

**Algorithm 2:** MM-Based Graph Laplacian Estimation (GLE-MM).

**Require:** Initialization: $\mathbf{w}^{(0)} \geq \mathbf{0}$, $l = 0$
1:  $\mathbf{R} = \mathbf{E}^T\mathbf{K}\mathbf{E}$;
2:  **repeat**
3:    $\mathbf{Q} = \mathrm{Diag}([\mathbf{w}^{(l)T}, 1/N]^T)\mathbf{G}^T\Big(\mathbf{G} \cdot \mathrm{Diag}([\mathbf{w}^{(l)T}, 1/N]^T)\mathbf{G}^T\Big)^{-1}\mathbf{G}\mathrm{Diag}([\mathbf{w}^{(l)T}, 1/N]^T)$;
4:    $\mathbf{Q}_M = \mathbf{Q}_{1:M, 1:M}$;
5:    $\mathbf{w}^{(l+1)} = \sqrt{\mathrm{diag}(\mathbf{Q}_M) \odot \mathrm{diag}(\mathbf{R})^{-1}}$;
6:    $l = l + 1$;
7:  **until** convergence

---

yet. For the sake of algorithmic simplicity, we need to further majorize the objective of (38). Thus, we introduce the following supporting lemma.

*Lemma 4 ([23]):* For any $\mathbf{YXY}^T \succ \mathbf{0}$, the following matrix inequality holds:

$$\left(\mathbf{YXY}^T\right)^{-1} \preceq \mathbf{Z}_0^{-1}\mathbf{YX}_0\mathbf{X}^{-1}\mathbf{X}_0\mathbf{Y}^T\mathbf{Z}_0^{-1}, \qquad (39)$$

where $\mathbf{Z}_0 = \mathbf{YX}_0\mathbf{Y}^T$. Equality is achieved at $\mathbf{X} = \mathbf{X}_0$.

As a result, we are able to do the following (define $\tilde{\mathbf{w}} \triangleq [\mathbf{w}^T, 1/N]^T$ and $\tilde{\mathbf{w}}_0 \triangleq [\mathbf{w}_0^T, 1/N]^T$):

$$\begin{aligned} &\mathrm{Tr}\left(\mathbf{F}_0\left(\mathbf{G}\mathrm{Diag}(\tilde{\mathbf{w}})\mathbf{G}^T\right)^{-1}\right) \\ &= \mathrm{Tr}\left(\mathbf{F}_0^{1/2}\left(\mathbf{G}\mathrm{Diag}(\tilde{\mathbf{w}})\mathbf{G}^T\right)^{-1}\mathbf{F}_0^{1/2}\right) \\ &\overset{(a)}{\leq} \mathrm{Tr}\left(\mathbf{F}_0^{1/2}\mathbf{F}_0^{-1}\mathbf{G}\mathrm{Diag}(\tilde{\mathbf{w}}_0)\mathrm{Diag}(\tilde{\mathbf{w}})^{-1}\right. \\ &\quad \left.\mathrm{Diag}(\tilde{\mathbf{w}}_0)\mathbf{G}^T\mathbf{F}_0^{-1}\mathbf{F}_0^{1/2}\right), \end{aligned} \qquad (40)$$

where (a) comes from (39) with $\mathbf{Y} = G$, $\mathbf{X} = \mathrm{diag}(\mathbf{w})$, $\mathbf{X}_0 = \mathrm{diag}(\tilde{\mathbf{w}}_0)$, $\mathbf{Z}_0 = \mathbf{F}_0$. The surrogate functions obtained in (37) and (40) are the majorizing functions of (33) that satisfy the assumptions A1−A4. To this point, the minimization problem is written as

$$\underset{\mathbf{w}\geq\mathbf{0}}{\text{minimize}} \quad \mathrm{diag}(\mathbf{R})^T\mathbf{w} + \mathrm{diag}(\mathbf{Q}_M)^T\mathbf{w}^{-1}, \qquad (41)$$

where $\mathbf{R} = \mathbf{E}^T\mathbf{K}\mathbf{E}$, $\mathbf{Q}_M = \mathbf{Q}_{1:M,1:M}$, and $\mathbf{Q} = \mathrm{Diag}(\tilde{\mathbf{w}}_0)\mathbf{G}^T(\mathbf{G}\mathrm{Diag}(\tilde{\mathbf{w}}_0)\mathbf{G}^T)^{-1}\mathbf{G}\mathrm{Diag}(\tilde{\mathbf{w}}_0)$. The optimal solution to (41) is

$$\mathbf{w}^\star = \sqrt{\mathrm{diag}(\mathbf{Q}_M) \odot \mathrm{diag}(\mathbf{R})^{-1}}. \qquad (42)$$

We summarize the MM solution in Algorithm 2.

### C. Computational Complexity

Analogously, we present the complexity analysis of Algorithm 2 as follows. Obviously, the most costly step is to compute the matrix $\mathbf{Q}$. When we have $\mathbf{Q}$, it takes $\mathcal{O}(M^2)$ to obtain $\mathbf{Q}_M$ and $\mathcal{O}(M)$ to get $\mathbf{w}^{(l+1)}$. It takes four mini-steps to compute $\mathbf{Q}$: i) matrix multiplication $\mathbf{G}\mathrm{Diag}([\mathbf{w}^{(l)T}, 1/N]^T)\mathbf{G}^T$: $\mathcal{O}(M^2N + N^2M)$; ii) matrix inversion: $\mathcal{O}(N^3)$; iii) matrix multiplication $\mathbf{G}\mathrm{Diag}([\mathbf{w}^{(l)T}, 1/N]^T)$: $\mathcal{O}(NM^2)$; and iv) matrix multiplication to obtain $\mathbf{Q}$: $\mathcal{O}(N^2M + NM^2)$. The overall complexity to get $\mathbf{Q}$ is $\mathcal{O}(M^2N + N^2M + N^3)$, as is the

per-iteration complexity of Algorithm 2, resulting from the five matrix multiplications and one matrix inversion. If the graph is almost complete, then $M = \mathcal{O}(N^2)$, resulting in an $\mathcal{O}(N^5)$ per-iteration cost. If the graph is sparse, then $M = \mathcal{O}(N)$, resulting in an $\mathcal{O}(N^3)$ per-iteration cost.

## V. GRAPH LAPLACIAN ESTIMATION WITH NOMINAL EIGENSUBSPACE

The estimation of the graph Laplacian requires a sample set $\{\mathbf{x}_i\}_{i=1}^T$. When the sample size $T$ is small, the sample covariance matrix $\mathbf{S}$ will be highly inaccurate, which hinders the estimation performance of the Laplacian. One possible way to improve the performance is to exploit some of the leading eigenvectors of the sample covariance matrix $\mathbf{S}$ as a reference of the true eigensubspace of $\mathbf{\Theta}$ (since $\mathbf{S}$ and $\mathbf{\Theta}$ share the same eigenvectors). Suppose we take into account $K(< N)$ leading eigenvectors (corresponding to $K$ largest eigenvalues) of $\mathbf{S}$, and thus the nominal eigensubspace is represented by $K$ orthogonal eigenvectors, denoted as $\hat{\mathbf{U}}_K \in \mathbb{R}^{N \times K}$, subject to inaccuracy caused by the limited number of samples. The nominal value of $\mathbf{\Theta}$, denoted as $\hat{\mathbf{\Theta}}$, can be expressed as

$$\hat{\mathbf{\Theta}} = \hat{\mathbf{U}}_K \mathbf{\Lambda}_K \hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp} \hat{\mathbf{U}}_{K\perp}^T, \qquad (43)$$

where $\mathbf{\Lambda}_K \in \mathbb{R}^{K \times K}$ is PSD and diagonal, $\mathbf{\Xi}_{K\perp} \in \mathbb{R}^{(N-K) \times (N-K)}$ is PSD (not necessarily diagonal), and $\hat{\mathbf{U}}_{K\perp}$ is the orthogonal complement of $\hat{\mathbf{U}}_K$, i.e., $\hat{\mathbf{U}}_{K\perp}^T \hat{\mathbf{U}}_{K\perp} = \mathbf{I}$ and $\hat{\mathbf{U}}_K^T \hat{\mathbf{U}}_{K\perp} = \mathbf{0}$. Here, $\mathbf{\Xi}_{K\perp}$ is not diagonal since we do not know the complement space defined by $\hat{\mathbf{U}}_{K\perp}$ exactly. Without diagonal limitations, the representation abilities of $\hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp} \hat{\mathbf{U}}_{K\perp}^T$ are stronger. The graph Laplacian estimation problem is recast as

$$\begin{aligned} \underset{\mathbf{\Theta}, \hat{\mathbf{\Theta}}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}}{\text{minimize}} \quad & \mathrm{Tr}(\mathbf{\Theta}\mathbf{K}) - \log\det(\mathbf{\Theta} + \mathbf{J}) \\ \text{subject to} \quad & \mathbf{\Theta} \in \mathcal{L}(\mathbf{A}) \\ & \hat{\mathbf{\Theta}} = \hat{\mathbf{U}}_K \mathbf{\Lambda}_K \hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp} \hat{\mathbf{U}}_{K\perp}^T \\ & \mathbf{\Lambda}_K = \mathrm{Diag}\left(\{\lambda_i \geq 0\}_{i=1}^K\right), \mathbf{\Xi}_{K\perp} \succeq \mathbf{0} \\ & \left\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}\right\|_F \leq \epsilon. \end{aligned} \qquad (44)$$

The last constraint controls the level of uncertainty, measured by the Frobenius norm.

*Remark 2:* In a recent work, Segarra *et al.* [30] proposed a framework that utilizes the eigenspace obtained from the second-order data statistics for the network inference, where, the objective is to estimate the topology from the stationary signals, assuming these to be generated from some diffusion process over a network, and due to the stationary property, the eigenspace of the data statistics (e.g., covariance matrix) is the same as the eigenspace of the network (also known as a graph shift operator). This property allows to utilize the eigenspace obtained from the data statistics for topology estimation. The proposed approach here does not assume any diffusion process and directly specializes in the estimation of the precision matrix as the target graph Laplacian. In this regard, the aim is to use the nominal eigenspace obtained from naive estimation of data statistics (sample covariance matrix) to refine the final estimation of the target matrix (graph Laplacian).

### A. The ADMM Approach

We can use the ADMM framework to solve (44). We apply the same reformulation method as Section III and obtain

$$\begin{aligned} \underset{\mathbf{\Xi}, \mathbf{C}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}, \mathbf{\Delta}}{\text{minimize}} \quad & \mathrm{Tr}\left(\mathbf{\Xi}\tilde{\mathbf{K}}\right) - \log\det(\mathbf{\Xi}) \\ \text{subject to} \quad & \mathbf{\Xi} \succeq \mathbf{0}, \ \mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C} = \mathbf{0} \\ & \left.\begin{array}{l} \mathbf{I} \odot \mathbf{C} \geq \mathbf{0} \\ \mathbf{B} \odot \mathbf{C} = \mathbf{0} \\ \mathbf{A} \odot \mathbf{C} \leq \mathbf{0} \end{array}\right\} \mathbf{C} \in \mathcal{C} \\ & \mathbf{P}\mathbf{\Xi}\mathbf{P}^T = \hat{\mathbf{U}}_K \mathbf{\Lambda}_K \hat{\mathbf{U}}_K^T + \\ & \hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp} \hat{\mathbf{U}}_{K\perp}^T + \mathbf{\Delta} \\ & \mathbf{\Lambda}_K = \mathrm{Diag}\left(\{\lambda_i \geq 0\}_{i=1}^K\right) \\ & \mathbf{\Xi}_{K\perp} \succeq \mathbf{0}, \ \|\mathbf{\Delta}\|_F \leq \epsilon, \end{aligned} \qquad (45)$$

where $\tilde{\mathbf{K}} = \mathbf{P}^T \mathbf{K} \mathbf{P}$ and $\mathbf{P}$ is the orthogonal complement of $\mathbf{1}$. The (partial) augmented Lagrangian is

$$\begin{aligned} & \mathcal{L}(\mathbf{\Xi}, \mathbf{C}, \mathbf{Y}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}, \mathbf{\Delta}, \mathbf{Z}) = \\ & \mathrm{Tr}\left(\mathbf{\Xi}\tilde{\mathbf{K}}\right) - \log\det(\mathbf{\Xi}) + \mathrm{Tr}\left(\mathbf{Y}^T\left(\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C}\right)\right) \\ & + \frac{\rho}{2}\left\|\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C}\right\|_F^2 + \mathrm{Tr}\Big(\mathbf{Z}^T\Big(\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \\ & \left(\hat{\mathbf{U}}_K \mathbf{\Lambda}_K \hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp} \hat{\mathbf{U}}_{K\perp}^T + \mathbf{\Delta}\right)\Big)\Big) \\ & + \frac{\rho}{2}\left\|\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \left(\hat{\mathbf{U}}_K \mathbf{\Lambda}_K \hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp} \hat{\mathbf{U}}_{K\perp}^T + \mathbf{\Delta}\right)\right\|_F^2. \end{aligned} \qquad (46)$$

We treat $\mathbf{\Xi}, \mathbf{C}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}$, and $\mathbf{\Delta}$ as primal variables and define $\mathbf{Y}$ and $\mathbf{Z}$ as the dual variables with respect to the constraints $\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C} = \mathbf{0}$ and $\mathbf{P}\mathbf{\Xi}\mathbf{P}^T = \hat{\mathbf{U}}_K \mathbf{\Lambda}_K \hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp} \hat{\mathbf{U}}_{K\perp}^T + \mathbf{\Delta}$, respectively. The other constraints are treated as implicit constraints. We separate the primal variables into three blocks: i) $\mathbf{\Xi}$; ii) $\mathbf{C}, \mathbf{\Lambda}_K$ and $\mathbf{\Xi}_{K\perp}$; and iii) $\mathbf{\Delta}$. A 3-block ADMM framework enjoys a convergence guarantee when the random permutation update rule is adopted; i.e., the update order of the three blocks is controlled by a random seed in every iteration [31]–[33]. Due to the randomization update mechanism, we omit the superscript "+" of the other variables in the primal update steps.

*1) Update of $\mathbf{\Xi}$:*

$$\mathbf{\Xi}^+ = \arg\min_{\mathbf{\Xi} \succeq \mathbf{0}} \mathcal{L}(\mathbf{\Xi}, \mathbf{C}, \mathbf{Y}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}, \mathbf{\Delta}, \mathbf{Z}). \qquad (47)$$

Let $\hat{\mathbf{\Theta}} = \hat{\mathbf{U}}_K \mathbf{\Lambda}_K \hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp} \hat{\mathbf{U}}_{K\perp}^T + \mathbf{\Delta}$, and we have

$$\begin{aligned} \mathbf{\Xi}^+ &= \arg\min_{\mathbf{\Xi} \succeq \mathbf{0}} \mathcal{L}(\mathbf{\Xi}, \mathbf{C}, \mathbf{Y}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}, \mathbf{\Delta}, \mathbf{Z}) \\ &= \mathrm{Tr}\left(\mathbf{\Xi}\tilde{\mathbf{K}}\right) - \log\det(\mathbf{\Xi}) + \mathrm{Tr}\left(\mathbf{P}^T\mathbf{Y}^T\mathbf{P}\mathbf{\Xi}\right) \\ &\quad + \frac{\rho}{2}\left\|\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{C}\right\|_F^2 + \mathrm{Tr}\left(\mathbf{P}^T\mathbf{Z}^T\mathbf{P}\mathbf{\Xi}\right) \\ &\quad + \frac{\rho}{2}\left\|\mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \hat{\mathbf{\Theta}}\right\|_F^2 \\ &\overset{(a)}{=} \rho\left\|\mathbf{\Xi} + \frac{1}{2\rho}\left(\tilde{\mathbf{K}} + \tilde{\mathbf{Y}} + \tilde{\mathbf{Z}}\right) - \frac{1}{2}\left(\tilde{\mathbf{C}} + \mathbf{P}^T\hat{\mathbf{\Theta}}\mathbf{P}\right)\right\|_F^2 \\ &\quad - \log\det(\mathbf{\Xi}) \\ &\overset{(b)}{=} \mathbf{U}\mathbf{D}\mathbf{U}^T, \end{aligned} \qquad (48)$$

where (a) $\tilde{\mathbf{Y}} = \mathbf{P}^T\mathbf{Y}\mathbf{P}$, $\tilde{\mathbf{Z}} = \mathbf{P}^T\mathbf{Z}\mathbf{P}$, and $\mathbf{C} = \mathbf{P}^T\mathbf{C}\mathbf{P}$ and (b) $\mathbf{U}$ comes from the eigenvalue decomposition of $\frac{1}{2\rho}(\tilde{\mathbf{S}} + \tilde{\mathbf{Y}} + \tilde{\mathbf{Z}}) - \frac{1}{2}(\tilde{\mathbf{C}} + \mathbf{P}^T\hat{\mathbf{\Theta}}\mathbf{P}) = \mathbf{P}^T\big(\frac{1}{2\rho}(\mathbf{S} + \mathbf{Y} + \mathbf{Z}) - \frac{1}{2}(\mathbf{C} + \hat{\mathbf{\Theta}})\big)\mathbf{P}$, i.e., $\mathbf{P}^T\big(\frac{1}{2\rho}(\mathbf{S} + \mathbf{Y} + \mathbf{Z}) - \frac{1}{2}(\mathbf{C} + \hat{\mathbf{\Theta}})\big)\mathbf{P} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, and $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \frac{-\rho\Lambda_{ii} + \sqrt{\rho^2\Lambda_{ii}^2 + 2\rho}}{2\rho}$ (see Lemma 1).

*2) Update of $\mathbf{C}$, $\mathbf{\Lambda}_K$, and $\mathbf{\Xi}_{K\perp}$:*

$$\left[\mathbf{C}^+, \mathbf{\Lambda}_K^+, \mathbf{\Xi}_{K\perp}^+\right]$$
$$= \arg\min_{\mathbf{C}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}} \mathcal{L}\left(\mathbf{\Xi}, \mathbf{C}, \mathbf{Y}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}, \mathbf{\Delta}, \mathbf{Z}\right). \quad (49)$$

As can be observed from the augmented Lagrangian, $\mathbf{C}$ and $[\mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}]$ are separated by summation. So we can apply Lemma 2:

$$\mathbf{C}^+ = \mathbf{I} \odot \left[\frac{1}{\rho}\mathbf{Y} + \mathbf{\Theta}\right]_+ + \mathbf{A} \odot \left[\frac{1}{\rho}\mathbf{Y} + \mathbf{\Theta}\right]_-, \quad (50)$$

with $\mathbf{\Theta} = \mathbf{P}\mathbf{\Xi}\mathbf{P}^T$. Meanwhile,

$$\left[\mathbf{\Lambda}_K^+, \mathbf{\Xi}_{K\perp}^+\right]$$
$$\overset{(a)}{=} \arg\min_{\mathbf{\Lambda}_K = \text{Diag}\left(\{\lambda_i \geq 0\}_{i=1}^K\right), \mathbf{\Xi}_{K\perp} \succeq \mathbf{0}}$$
$$- \text{Tr}\left(\mathbf{Z}^T\left(\hat{\mathbf{U}}_K\mathbf{\Lambda}_K\hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp}\mathbf{\Xi}_{K\perp}\hat{\mathbf{U}}_{K\perp}^T\right)\right)$$
$$+ \frac{\rho}{2}\left\|\hat{\mathbf{\Theta}} - \left(\hat{\mathbf{U}}_K\mathbf{\Lambda}_K\hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp}\mathbf{\Xi}_{K\perp}\hat{\mathbf{U}}_{K\perp}^T\right)\right\|_F^2 \quad (51)$$
$$\overset{(b)}{=} \begin{cases} \arg\min_{\mathbf{\Lambda}_K = \text{Diag}\left(\{\lambda_i \geq 0\}_{i=1}^K\right)} -\text{Tr}\left(\mathbf{Z}_K^T\mathbf{\Lambda}_K\right) \\ \quad + \frac{\rho}{2}\|\mathbf{\Lambda}_K - \mathbf{W}_K\|_F^2 \\ \arg\min_{\mathbf{\Xi}_{K\perp} \succeq \mathbf{0}} -\text{Tr}\left(\mathbf{Z}_{K\perp}^T\mathbf{\Xi}_{K\perp}\right) \\ \quad + \frac{\rho}{2}\|\mathbf{\Xi}_{K\perp} - \mathbf{W}_{K\perp}\|_F^2 \end{cases}$$
$$= \begin{cases} [W_{K,ii} + Z_{K,ii}/\rho]_+ \quad \forall i = 1, \ldots, K \\ [\mathbf{W}_{K\perp} + \mathbf{Z}_{K\perp}/\rho]_{\text{PSD}}, \end{cases}$$

where (a) $\hat{\mathbf{\Theta}} = \mathbf{P}\mathbf{\Xi}\mathbf{P}^T - \mathbf{\Delta}$ and (b) $\mathbf{Z}_K = \hat{\mathbf{U}}_K^T\mathbf{Z}\hat{\mathbf{U}}_K$, $\mathbf{W}_K = \hat{\mathbf{U}}_K^T\hat{\mathbf{\Theta}}\hat{\mathbf{U}}_K$, $\mathbf{Z}_{K\perp} = \hat{\mathbf{U}}_{K\perp}^T\mathbf{Z}\hat{\mathbf{U}}_{K\perp}$, and $\mathbf{W}_{K\perp} = \hat{\mathbf{U}}_{K\perp}^T\hat{\mathbf{\Theta}}\hat{\mathbf{U}}_{K\perp}$.

*3) Update of $\mathbf{\Delta}$:*

$$\mathbf{\Delta}^+ = \arg\min_{\|\mathbf{\Delta}\|_F \leq \epsilon} \mathcal{L}\left(\mathbf{\Xi}, \mathbf{C}, \mathbf{Y}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp}, \mathbf{\Delta}, \mathbf{Z}\right). \quad (52)$$

Let $\hat{\mathbf{\Delta}} = \mathbf{P}\mathbf{\Xi}\mathbf{P}^T - (\hat{\mathbf{U}}_K\mathbf{\Lambda}_K\hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp}\mathbf{\Xi}_{K\perp}\hat{\mathbf{U}}_{K\perp}^T)$, and we have

$$\mathbf{\Delta}^+ = \arg\min_{\|\mathbf{\Delta}\|_F \leq \epsilon} -\text{Tr}\left(\mathbf{Z}^T\mathbf{\Delta}\right) + \frac{\rho}{2}\left\|\hat{\mathbf{\Delta}} - \mathbf{\Delta}\right\|_F^2$$
$$= \arg\min_{\|\mathbf{\Delta}\|_F \leq \epsilon} \frac{\rho}{2}\left\|\mathbf{\Delta} - \left(\frac{1}{\rho}\mathbf{Z} + \hat{\mathbf{\Delta}}\right)\right\|_F^2$$
$$= \min\left(\frac{\epsilon}{\left\|\frac{1}{\rho}\mathbf{Z} + \hat{\mathbf{\Delta}}\right\|_F}, 1\right)\left(\frac{1}{\rho}\mathbf{Z} + \hat{\mathbf{\Delta}}\right). \quad (53)$$

The last step is to update the dual variables:

$$\mathbf{Y}^+ = \mathbf{Y} + \rho\left(\mathbf{\Theta}^+ - \mathbf{C}^+\right) \quad (54)$$

and

$$\mathbf{Z}^+ = \mathbf{Z} + \rho\Big(\mathbf{\Theta}^+ - \Big(\hat{\mathbf{U}}_K\mathbf{\Lambda}_K^+\hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp}\mathbf{\Xi}_{K\perp}^+\hat{\mathbf{U}}_{K\perp}^T + \mathbf{\Delta}^+\Big)\Big),$$

with $\mathbf{\Theta}^+ = \mathbf{P}\mathbf{\Xi}^+\mathbf{P}^T$. We summarize all the aforementioned primal update steps as follows and present the whole procedure in Algorithm 3. Note that the update order of the primal variable blocks requires random permutation [31] for the sake of convergence.

**Update $\mathbf{\Xi}$**
$$\begin{cases} \hat{\mathbf{\Theta}} = \hat{\mathbf{U}}_K\mathbf{\Lambda}_K\hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp}\mathbf{\Xi}_{K\perp}\hat{\mathbf{U}}_{K\perp}^T + \mathbf{\Delta} \\ \text{Eigenvalue decomposition:} \\ \mathbf{P}^T\left(\frac{1}{2\rho}(\mathbf{S} + \mathbf{Y} + \mathbf{Z}) - \frac{1}{2}\left(\mathbf{C} + \hat{\mathbf{\Theta}}\right)\right)\mathbf{P} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \\ \mathbf{D} \text{ is diagonal with } D_{ii} = \frac{-\rho\Lambda_{ii} + \sqrt{\rho^2\Lambda_{ii}^2 + 2\rho}}{2\rho} \\ \mathbf{\Xi}^+ = \mathbf{U}\mathbf{D}\mathbf{U}^T \end{cases}$$
$$(55)$$

**Update C**
$$\begin{cases} \mathbf{\Theta} = \mathbf{P}\mathbf{\Xi}\mathbf{P}^T \\ \mathbf{C}^+ = \mathbf{I} \odot \left[\frac{1}{\rho}\mathbf{Y} + \mathbf{\Theta}\right]_+ \\ \quad + \mathbf{A} \odot \left[\frac{1}{\rho}\mathbf{Y} + \mathbf{\Theta}\right]_- \end{cases} \quad (56)$$

**Update $\mathbf{\Lambda}_K$ and $\mathbf{\Xi}_{K\perp}$**
$$\begin{cases} \hat{\mathbf{\Theta}} = \mathbf{\Theta} - \mathbf{\Delta} \\ \mathbf{Z}_K = \hat{\mathbf{U}}_K^T\mathbf{Z}\hat{\mathbf{U}}_K, \ \mathbf{W}_K = \hat{\mathbf{U}}_K^T\hat{\mathbf{\Theta}}\hat{\mathbf{U}}_K \\ \mathbf{Z}_{K\perp} = \hat{\mathbf{U}}_{K\perp}^T\mathbf{Z}\hat{\mathbf{U}}_{K\perp}, \ \mathbf{W}_{K\perp} = \hat{\mathbf{U}}_{K\perp}^T\hat{\mathbf{\Theta}}\hat{\mathbf{U}}_{K\perp} \\ \mathbf{\Lambda}_K^+ = \text{Diag}\left(\left\{[W_{K,ii} + Z_{K,ii}/\rho]_+\right\}_{i=1}^K\right) \\ \mathbf{\Xi}_{K\perp}^+ = [\mathbf{W}_{K\perp} + \mathbf{Z}_{K\perp}/\rho]_{\text{PSD}} \end{cases}$$
$$(57)$$

**Update $\mathbf{\Delta}$**
$$\begin{cases} \hat{\mathbf{\Delta}} = \mathbf{\Theta} - \left(\hat{\mathbf{U}}_K\mathbf{\Lambda}_K\hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp}\mathbf{\Xi}_{K\perp}\hat{\mathbf{U}}_{K\perp}^T\right) \\ \mathbf{\Delta}^+ = \min\left(\frac{\epsilon}{\left\|\frac{1}{\rho}\mathbf{Z} + \hat{\mathbf{\Delta}}\right\|_F}, 1\right)\left(\frac{1}{\rho}\mathbf{Z} + \hat{\mathbf{\Delta}}\right) \end{cases}$$
$$(58)$$

*Remark 3:* Algorithm 3 outlines the update step for the GLENE formulation with $K < N$, where $K$ is the number of reliable eigenvectors. However, in many applications, all the eigenvectors can be obtained reliably, and including all these reliable eigenvectors in the GLENE formulation (44) will yield a better estimation result. This can be easily accommodated in the current formulation by setting $K = N$, which also implies that the term involving the orthogonal complement is not required in the formulation $\hat{\mathbf{U}}_{K\perp}\mathbf{\Xi}_{K\perp}\hat{\mathbf{U}}_{K\perp}^T$. Further, this also reduces the computational burden of the algorithm, as now the variable $\hat{\mathbf{U}}_{K\perp}^T$ is not present, which will simplify the update in (57).

### B. Computational Complexity

We present the complexity analysis of Algorithm 3 as follows. In the primal update, we need to update $\mathbf{\Xi}$, $\mathbf{C}$, $\mathbf{\Lambda}_K$, $\mathbf{\Xi}_{K\perp}$, and $\mathbf{\Delta}$. For $\mathbf{\Xi}$ (see (55)), the costly steps are i) matrix multiplications to obtain $\hat{\mathbf{\Theta}}$: $\mathcal{O}(NK^2 + KN^2 + N(N - K)^2 + (N - K)N^2)$; ii) matrix multiplications $\mathbf{P}^T(\frac{1}{2\rho}(\mathbf{S} + \mathbf{Y} + \mathbf{Z}) - \frac{1}{2}(\mathbf{C} + \hat{\mathbf{\Theta}}))\mathbf{P}$: $\mathcal{O}(N^3)$; iii) eigenvalue decomposition: $\mathcal{O}(N^3)$; and iv) matrix multiplication $\mathbf{U}\mathbf{D}\mathbf{U}^T$: $\mathcal{O}(N^3)$. The overall cost is $\mathcal{O}(NK^2 + KN^2 + N(N - K)^2$

---

**Algorithm 3:** Graph Laplacian Estimation with Nominal Eigensubspace (GLENE).

---

**Require:** Initialization: $\mathbf{K}, \hat{\mathbf{U}}_K$, symmetric $\mathbf{Y}^{(0)}, \mathbf{Z}^{(0)}$, $\mathbf{C}^{(0)}$, and $\mathbf{\Delta}^{(0)}$ with $\|\mathbf{\Delta}^{(0)}\|_F \leq \epsilon$, diagonal nonnegative $\mathbf{\Lambda}_K^{(0)}$, positive semidefinite $\mathbf{\Xi}^{(0)}$ and $\mathbf{\Xi}_{K\perp}^{(0)}, \rho > 0, l = 0$

1:   **repeat**
2:     **Primal Update:** $[\mathbf{\Xi}^{(l)}, \mathbf{C}^{(l)}, \mathbf{\Lambda}_K^{(l)}, \mathbf{\Xi}_{K\perp}^{(l)}, \mathbf{\Delta}^{(l)}] \longrightarrow$
       $[\mathbf{\Xi}^{(l+1)}, \mathbf{C}^{(l+1)}, \mathbf{\Lambda}_K^{(l+1)}, \mathbf{\Xi}_{K\perp}^{(l+1)}, \mathbf{\Delta}^{(l+1)}]$

$\begin{cases} \text{Update } \mathbf{\Xi}, \text{ cf. (55);} \\ \text{Update } \mathbf{C}, \mathbf{\Lambda}_K, \mathbf{\Xi}_{K\perp} \text{ cf. (56) and (57);} \\ \text{Update } \mathbf{\Delta}, \text{ cf. (58);} \end{cases}$ $\begin{array}{c} \text{Randomized} \\ \text{update order} \end{array}$

3:     **Dual Update:** $[\mathbf{Y}^{(l)}, \mathbf{Z}^{(l)}] \longrightarrow [\mathbf{Y}^{(l+1)}, \mathbf{Z}^{(l+1)}]$

$\begin{cases} \mathbf{Y}^{(l+1)} = \mathbf{Y}^{(l)} + \rho(\mathbf{\Theta}^{(l+1)} - \mathbf{C}^{(l+1)}); \\ \mathbf{Z}^{(l+1)} = \mathbf{Z}^{(l)} + \rho\Big(\mathbf{\Theta}^{(l+1)} - \\ \quad (\hat{\mathbf{U}}_K \mathbf{\Lambda}_K^{(l+1)} \hat{\mathbf{U}}_K^T + \hat{\mathbf{U}}_{K\perp} \mathbf{\Xi}_{K\perp}^{(l+1)} \hat{\mathbf{U}}_{K\perp}^T + \mathbf{\Delta}^{(l+1)})\Big); \end{cases}$

4:     $l = l + 1;$
5:   **until** convergence

---



Fig. 1.    Optimality gap versus dimension $N$ and $M = 4N$.



Fig. 2.    Performance of naive estimation: estimation error versus $T/N$, $N = 30$.

$+ (N - K)N^2 + N^3$). For $\mathbf{C}$ (see (56)), the complexity is $\mathcal{O}(N^3)$, the same as in Section III-C. For $\mathbf{\Lambda}_K$ (see (57)), the costly steps are matrix multiplications $\hat{\mathbf{U}}_K^T \mathbf{Z} \hat{\mathbf{U}}_K$ and $\hat{\mathbf{U}}_K^T \hat{\mathbf{\Theta}} \hat{\mathbf{U}}_K$: $\mathcal{O}(NK^2 + KN^2)$ for both. For $\mathbf{\Xi}_{K\perp}$ (see (57)), the costly steps are i) matrix multiplications $\hat{\mathbf{U}}_{K\perp}^T \mathbf{Z} \hat{\mathbf{U}}_{K\perp}$ and $\hat{\mathbf{U}}_{K\perp}^T \hat{\mathbf{\Theta}} \hat{\mathbf{U}}_{K\perp}$: $\mathcal{O}(N(N - K)^2 + (N - K)N^2)$ for both, and ii) projection to the PSD cone: $\mathcal{O}((N - K)^3)$. For $\mathbf{\Delta}$ (see (58)), the costly step is merely the matrix multiplications to obtain $\hat{\mathbf{\Delta}}$: $\mathcal{O}(NK^2 + KN^2 + N(N - K)^2 + (N - K)N^2)$.
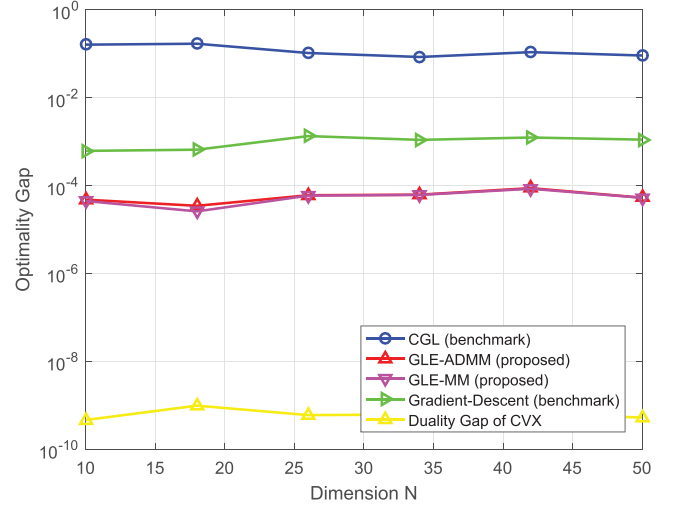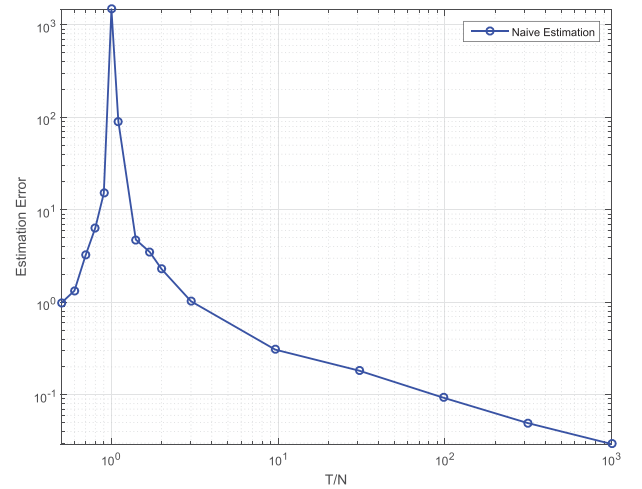
In dual update, the update of $\mathbf{Y}$ costs $\mathcal{O}(N^2)$ operations, while the cost of updating $\mathbf{Z}$ is $\mathcal{O}(NK^2 + KN^2 + N(N - K)^2 + (N - K)N^2)$ due to matrix multiplication operations. Since $K < N$, the per-iteration complexity is $\mathcal{O}(N^3)$, with twenty-six matrix multiplications, one eigenvalue decomposition, and one projection to the PSD cone.

## VI. NUMERICAL SIMULATIONS

In this section, we present numerical results for both synthetic and real-data experiments. All simulations are performed on a PC with a 3.20 GHz i5-4570 CPU and 8 GB RAM. The off-the-shelf solver is specified as MOSEK [34] built in the CVX toolbox [35]. The MOSEK solver itself does not support functions from the exponential family, e.g., exp and log, so we cannot bypass the CVX toolbox and call MOSEK directly. The proposed algorithms are terminated when the Frobenius norm of the change of $\mathbf{\Theta}$ between iterations is smaller than a threshold (by default $10^{-7}$) or the number of iterations reaches a predetermined maximum (by default $10^5$). The reported performance of any single data point comes from the average of 100 randomized instances (random connected graphs). The tuning parameters for the update of $\rho$ are $\rho^{(0)} = 1, \mu = 2$, and $\tau^{\text{inc}} = \tau^{\text{dec}} = 2$ by default.

### A. Synthetic Experiments — Graph Laplacian Estimation

*Experiment Settings:* We set the number of edges for true topology $M_{\text{true}} = 4N$ and generate a random adjacency matrix $\mathbf{A}_{\text{true}}$ corresponding to a connected graph. We generate a GMRF model parameterized by the true precision matrix $\mathbf{\Theta}_{\text{true}}$, which satisfies the aforementioned topology as well as the Laplacian constraints: $\mathbf{\Theta}_{\text{true}} \in \mathcal{L}(\mathbf{A}_{\text{true}})$. A total of $T$ samples $\{\mathbf{x}_i\}_{i=1}^T$ are drawn from this GMRF model: $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Theta}_{\text{true}}^\dagger), \forall i$. The sample covariance matrix $\mathbf{S}$ is computed as $\mathbf{S} = \frac{1}{T}\sum_{i=1}^T (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$, with $\bar{\mathbf{x}} = \frac{1}{T}\sum_{i=1}^T \mathbf{x}_i$. In the experiment, only $\{\mathbf{x}_i\}_{i=1}^T$ and $\mathbf{A}_{\text{true}}$ are provided for the estimation of $\mathbf{\Theta}_{\text{true}}$. The sparsity parameter $\alpha$ is set to be 0.005.

*1) Comparison of Optimality:* First, we present the simulation results on optimality. We set $T = 10N$. With $\mathbf{S}$ and $\mathbf{A}_{\text{true}}$, we can compare different algorithms for solving (12). We compare our proposed algorithms, i.e., Algorithm 1 GLE-ADMM and Algorithm 2 GLE-MM , with the benchmark algorithm in [1, Algorithm 2 CGL]. For the CGL algorithm we use the code provided by the authors.[2] The projected gradient descent algorithm is also included to solve the problem with the tuned

---

[2]https://github.com/STAC-USC/Graph_Learning

(a) $T = N/2$.  (b) $T = N$.  (c) $T = 10^3 N$.
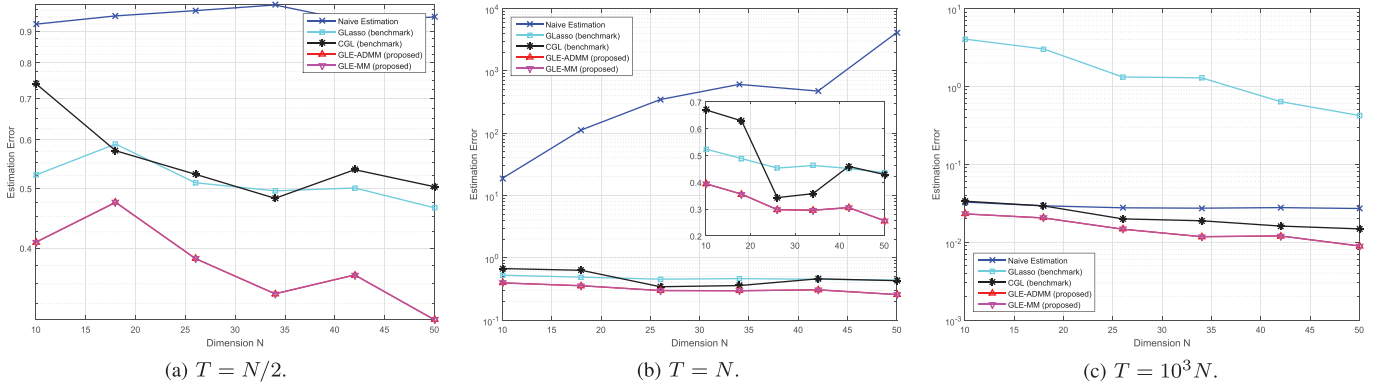
Fig. 3. Estimation Error versus dimension $N$ and $M = 4N$.

step size. Since all the algorithms are solving the same convex optimization problem, we compare their performances by benchmarking against the optimal solution to the optimization problem. The algorithmic performance is measured by the optimality gap, defined as $\|\mathbf{\Theta}_{\text{estimated}} - \mathbf{\Theta}^\star\|_F / \|\mathbf{\Theta}^\star\|_F$, with $\mathbf{\Theta}^\star$ as the optimal solution computed with CVX with enough iterations to achieve a duality gap of $10^{-10}$ (see yellow curve of Figure 1). On our simulation platform, the computational limit for CVX to solve (12) is $N = 50$. We show in Figure 1 that the proposed algorithms GLE-ADMM and GLE-MM can achieve optimality of $10^{-4}$, while the gap level of the benchmark CGL is around 0.1. The proposed algorithms are around three orders of magnitude more accurate than the benchmark.

*2) Comparison of Estimation Error:* The next step is to compare the estimation error, which is defined as $\|\mathbf{\Theta}_{\text{estimated}} - \mathbf{\Theta}_{\text{true}}\|_F / \|\mathbf{\Theta}_{\text{true}}\|_F$. One naive estimation of $\mathbf{\Theta}_{\text{true}}$ is $\mathbf{S}^\dagger$ (or $\mathbf{S}^{-1}$ if $\mathbf{S}$ is full rank). In Figure 2, we can see the performance of this trivial solution. When the sample size is small, e.g., $T = N/2$, the estimation error is close to 1; when the sample size $T$ equals $N$, the error level reaches a peak of over $10^3$; when the sample size is overwhelmingly large, e.g., $T = 10^3 N$, the estimation error goes down to $10^{-2}$. We focus on studying these three critical cases to see if the proposed GLE-ADMM and GLE-MM can perform better than the naive solution, CGL, and GLasso [14]. The sparsity parameter $\alpha$ is selected from $\{0\} \cup \{0.75^r(s_{\max}\sqrt{\log(N)/T})|r = 1, 2, \ldots, 14\}$, with $s_{\max} = \max_{i \neq j} |\mathbf{S}_{ij}|$ [1], and we choose the one that achieves the smallest estimation error. It can be observed in Figure 3 that the proposed GLE-ADMM and GLE-MM achieve the smallest estimation error across the entire range of $N$, whatever the sample size. For the small sample scenario, the proposed methods are significantly better than the benchmarks, with an improvement of 0.1 compared with the second lowest estimation error. For the equal sample scenario, the proposed methods improve the second lowest estimation error by at least 0.05. For the large sample scenario, the proposed methods narrowly beat the benchmarks, with an improvement of 0.01 in estimation error.

*3) Comparison of Computational Complexity:* Although the two proposed algorithms give almost the same optimality performance, it remains to be seen which one is more efficient. As was previously mentioned in Section III-C and IV-C, the per-iteration complexity of GLE-ADMM is $\mathcal{O}(N^3)$, while that of GLE-MM ranges from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^5)$, depending on
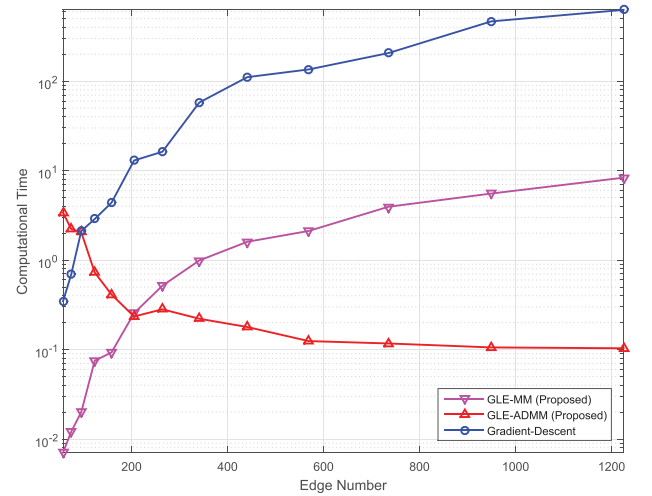


Fig. 4. Computational time (sec) versus edge number $M$, $N = 50$.

the number of nonzero elements in the adjacency matrix $\mathbf{A}$. We fix the number of nodes $N = 50$ and vary the edge number $M$ from $N + 7$ to $N(N-1)/2$. The comparison is presented in Figure 4. When the graph has a sparse topology, i.e., $M = \mathcal{O}(N)$, the computational time of GLE-MM is shorter than GLE-ADMM. For $M = N + 7$, GLE-MM is more than two orders of magnitude faster. However, when the graph is close to complete, i.e., $M = \mathcal{O}(N^2)$, GLE-ADMM is more efficient. For $M = N(N-1)/2$, GLE-ADMM is nearly two orders of magnitude faster. We can also observe that the two algorithms are equally efficient when $M \approx 200 = 4N$. The projected gradient descent algorithm holds a similar trend as GLE-MM since the sparsity of edges can benefit the computational time of it. But in any case, the performance of the projected gradient descent is worse than that of GLE-MM.

### B. Synthetic Experiments — Graph Laplacian Estimation With Nominal Eigensubspace

Experiment settings follow Section VI-A.

*1) Necessity of Nominal Eigensubspace:* We will show the necessity of considering a nominal eigensubspace with the following experiments. We set $N = 30$ and compute the eigenvectors of the sample covariance matrix $\mathbf{S}$, denoted as $\mathbf{U}$. We propose to estimate $\mathbf{\Theta}_{\text{true}}$ from solving (44). For a fixed sample size $T$, we choose $K$ leading eigenvectors (corresponding to
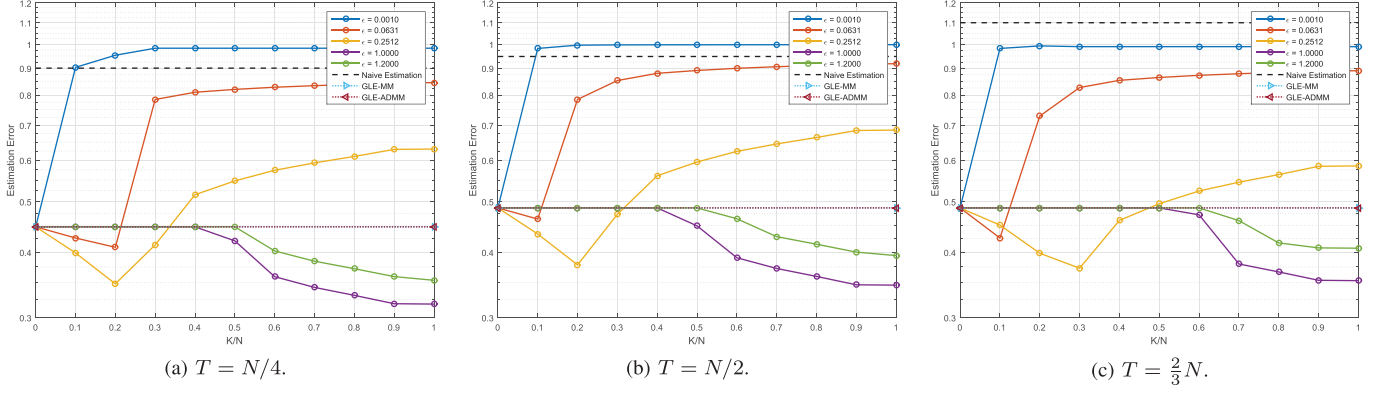
Fig. 5. Estimation error of estimated graph Laplacians: different choices of leading eigenvectors and uncertainty levels.

the $K$ largest eigenvalues) to help estimate the graph Laplacian, subject to different levels of uncertainty.

The simulation results are given in Figure 5. Whatever the sample size, a larger $\epsilon$ is always preferred for a low estimation error, consistent with the empirical results in [30, Sec. V-D]. When the sample size is smaller than dimension $N$ (number of nodes), a decreasing trend in the estimation error can be observed (though probably followed by an upward trend), as $K/N$ increases if $\epsilon$ is larger than 0.05. This decreasing trend indicates the necessity of including a nominal eigensubspace. If $K/N = 0$, problem (44) degenerates to problem (12). For the small sample scenario, the smallest estimation error is achieved when $\epsilon = 1$ and $K = N$, and it is approximately $1/3$ the error of the naive solution and $7/10$ that of the $K = 0$ scenario. When the sample size is equal to $N$, the curves behave similarly. The only difference is the quantity of improvement: for the equal sample scenario, the smallest estimation error is at least three orders of magnitude lower than that of the naive solution and approximately $7/10$ the error of the $K = 0$ scenario. For a sufficiently large $\epsilon$ (in this case 0.1), it makes no difference whether we consider a nominal eigensubspace, as is implied by the flat light-blue line.

*2) Optimality Concerns:* Now that the necessity of the nominal eigensubspace is justified, we look into the optimality performance of Algorithm 3 GLENE. We study the optimality gap of GLENE with respect to the CVX toolbox. The simulation results are given in Figure 6. We can see that the optimality gap is less than $10^{-4}$ across the entire range of $N$, indicating the good optimality performance of GLENE.

### C. Real-Data Experiments — Correlation of Stocks

We apply the aforementioned graph learning techniques to study pairwise correlations between a certain number of stocks. The stock pool consists of 30 stocks (listed in Table I), randomly drawn from the components of the S&P 100 Index, with a trading period from Apr. 1st, 2006 to Dec. 31st, 2015. Our objective is to reveal the strong correlations among these stocks and to filter out the weak correlations. For comparison, we will present the results of GLasso [14] as well. GLasso does not consider the Laplacian constraints or the nominal eigensubspace, and only
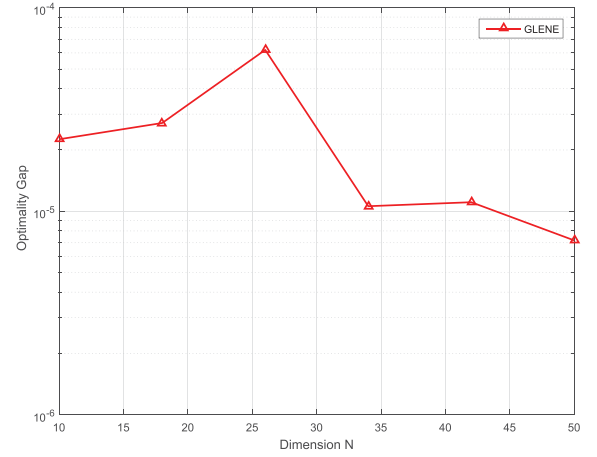


Fig. 6. Optimality gap versus dimension $N$, $M = 4N$, $K/N = 0.4$, and $\epsilon = 0.1$.

TABLE I
LIST OF STOCK POOL. (1. CONSUMER GOODS, 2. HEALTHCARE, 3. TECHNOLOGY, 4. FINANCIAL, 5. SERVICES, 6. INDUSTRIAL GOODS, 7. BASIC MATERIALS, 8. UTILITIES)

| Stock | Sector | Stock | Sector | Stock | Sector |
|-------|--------|-------|--------|-------|--------|
| AAPL | 1 | UNH | 2 | TWX | 5 |
| F | 1 | INTC | 3 | UNP | 5 |
| MO | 1 | AXP | 4 | DHR | 6 |
| NKE | 1 | GS | 4 | GE | 6 |
| PG | 1 | MET | 4 | MMM | 6 |
| BMY | 2 | USB | 4 | COP | 7 |
| GILD | 2 | AMZN | 5 | CVX | 7 |
| JNJ | 2 | DIS | 5 | OXY | 7 |
| LLY | 2 | MCD | 5 | SLB | 7 |
| MDT | 2 | PCLN | 5 | NEE | 8 |

requires positive semidefiniteness. We set $\mathbf{A} = \mathbf{1}\mathbf{1}^T - \mathbf{I}$ to indicate that there is no predefined graph topology (i.e., all the nodes are connected to each other) and $\epsilon = 0.1$. For the large sample scenario, the eigenspace of the sample covariance matrix should be reliable; the choice of $\epsilon$ is inferred from previous simulation results. The performance is evaluated in terms of i) sparsity ratio, $\sum_{i<j} I\{|\Theta_{ij}| < 10^{-3}\}/[N(N-1)/2]$, and ii) strong correlation ratio, $\sum_{i<j} I\{|\Theta_{ij}| > 0.3\}/[N(N-1)/2]$. For fair comparison, all the precision matrices or graph Laplacian matrices are diagonally normalized: $\text{Ddiag}(\mathbf{\Theta})^{-1/2}\mathbf{\Theta}\text{Ddiag}(\mathbf{\Theta})^{-1/2}$.
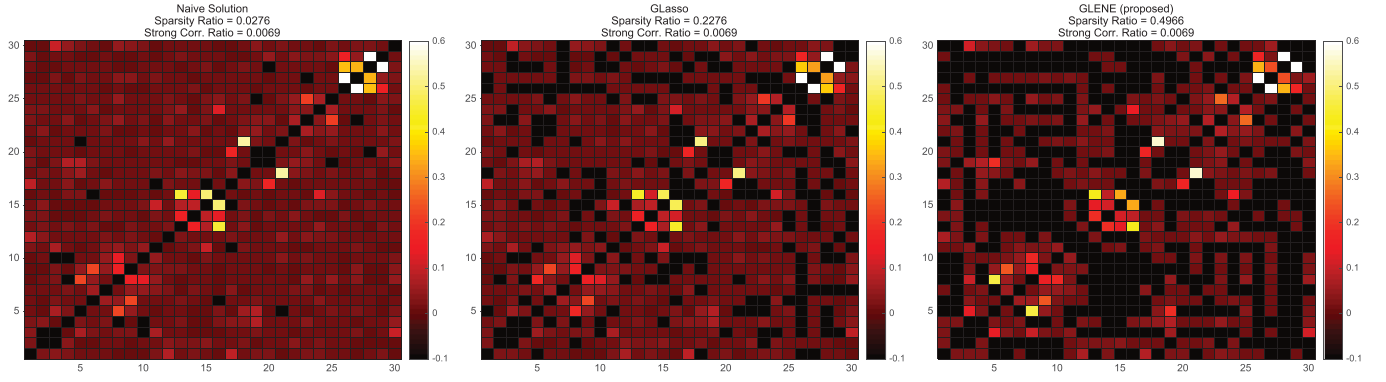
Fig. 7. Stock correlation visualization (principal diagonal values are removed).

TABLE II
A DETAILED COMPARISON OF GLENE AND GLASSO UNDER DIFFERENT CASES OF WELL-TUNED PARAMETERS

| | Sparsity ratio | Strong correlation ratio |
|---|---|---|
| GLasso, $\alpha = 0.02$ | 0.2276 | 0.0069 |
| GLE-MM, $\alpha = 0.02$ | 0.4121 | 0.0078 |
| GLE-ADMM, $\alpha = 0$ | 0.4121 | 0.0078 |
| GLENE, $\alpha = 0$, $K = 5$ | 0.5287 | 0.0092 |
| GLENE, $\alpha = 0$, $K = 10$ | 0.4529 | 0.0138 |
| GLENE, $\epsilon = 1.0512$, $K = 30$ | 0.5453 | 0.0291 |

First, we plot the correlations obtained from the optimized solutions of various algorithms, as is shown in Figure 7 (principal diagonal values are removed). We perform the following nonlinear transform on $\Theta_{ij}$ for better visualization: $f(x) = \{ \begin{smallmatrix} -0.1 & |x|<10^{-3} \\ 1/(1+\exp(-15(x-0.3))) & |x|>10^{-3} \end{smallmatrix}$. The number of leading eigenvectors $K$ is set to be 1, and the sparsity parameter $\alpha$ is chosen as 0.02 for the moment. Typically, the covariance of the stock returns has just one strong eigenvalue and we assume that eigenvector is well estimated. The naive solution refers to the pseudo-inverse or inverse of the sample covariance matrix. We can see that the proposed GLENE enjoys the largest sparsity ratio of 0.4966, much higher than that of the classical GLasso, 0.2276. As for the strong correlation ratio, the three methods give the same performance of 0.0069. Moreover, we find that the strong correlations cluster along the principal diagonal and the pattern exhibits a blockwise structure. This is because stocks of the same sector are more strongly correlated than those from a different one.

Next, we will take a closer look at the two methods GLasso and GLENE. GLENE has two tuning parameters, $\alpha$ and $K$, while GLasso only needs $\alpha$. We set $\alpha$ to be 0 (no sparsity promotion), 0.02 (low sparsity promotion), and 1 (high sparsity promotion). The simulation results are given in Figure 8. In order to achieve a tradeoff between the sparsity ratio and strong correlation ratio, the best parameter $\alpha$ for GLasso is 0.02, and the best parameter $\alpha$ for GLENE is 0. As for the optimal choice of $K$, $K = N$ yields the best performance, namely, the highest sparsity ratio and the highest strong correlation ratio. For the choice of $K < N$, there are two candidates: $K = 5$ (highest sparsity ratio) and $K = 10$ (highest strong correlation ratio), respectively. Table II lists a detailed comparison of GLasso and GLENE under a few cases of well-tuned parameters. We can see that for either choice of $K$,



(a) Sparsity ratio.
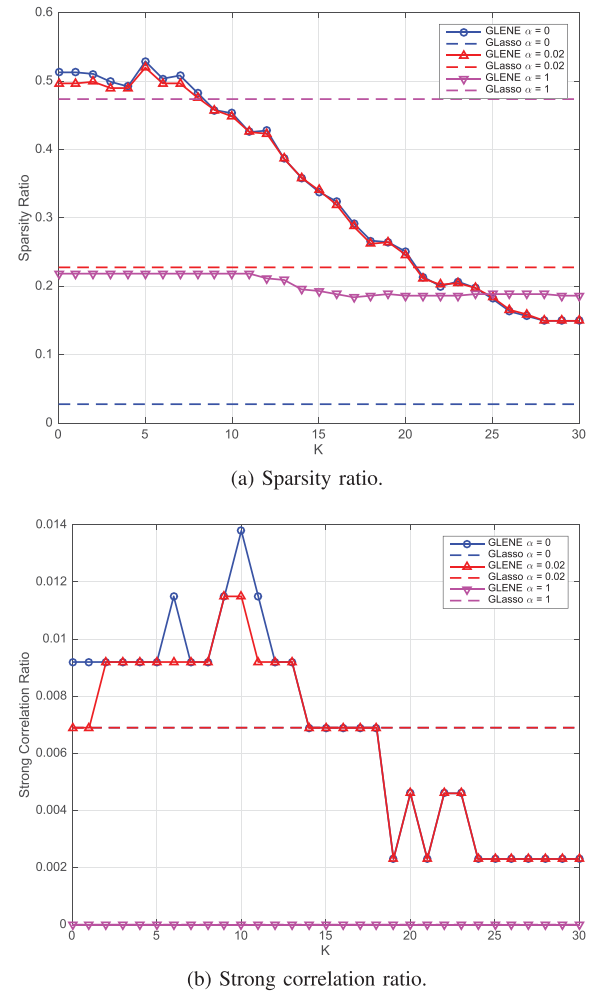


(b) Strong correlation ratio.

Fig. 8. Sparsity ratio and strong correlation ratio comparison: GLENE and GLasso.

GLENE enjoys a higher sparsity ratio and stronger correlation ratio than GLasso, indicating a better capability of structural exploration. We present in Table III the strongly correlated stocks indicated by GLENE. It can be observed that we have detected a strong correlation between two stocks of different sectors: PG (from Consumer Goods) and JNJ (from Healthcare). They both provide a massive array of healthcare products and they are competitors to each other.

TABLE III
STRONGLY CORRELATED STOCKS FOR DIFFERENT CHOICES OF $K$ (OBTAINED FROM GLENE). (1: CONSUMER GOODS, 2: HEALTHCARE, 3: TECHNOLOGY, 4: FINANCIAL, 5: SERVICES, 6: INDUSTRIAL GOODS, 7: BASIC MATERIALS, 8: UTILITIES)

| $K = 5$ | | | $K = 10$ | | |
|---|---|---|---|---|---|
| Stock pair | Sectors | Correlation | Stock pair | Sectors | Correlation |
| COP-CVX | 7-7 | 0.4866 | COP-CVX | 7-7 | 0.4546 |
| DIS-TWX | 5-5 | 0.3617 | BMY-LLY | 2-2 | 0.3833 |
| PG-JNJ | 1-2 | 0.3484 | DIS-TWX | 5-5 | 0.3718 |
| OXY-SLB | 7-7 | 0.3325 | DHR-MMM | 6-6 | 0.3162 |
| | | | OXY-SLB | 7-7 | 0.3033 |
| | | | JNJ-MDT | 2-2 | 0.3023 |

TABLE IV
COMPARISON OF GLE-MM, GLE-ADMM AND CGL FOR LYMPH NODE STATUS DATA

| Lymph data | Objective Value | Optimality Gap | CPU Time (s) |
|---|---|---|---|
| CGL | 154.93 | 1.92 | 55 |
| GLE-MM | 154.11 | 130e-04 | 153 |
| GLE-ADMM | 154.11 | 1.28e-04 | 79 |

TABLE V
COMPARISON OF GLE-MM, GLE-ADMM AND CGL FOR ARABIDOPSIS THALIANA DATA

| Arabidopsis | Objective Value | Optimality Gap | CPU Time (s) |
|---|---|---|---|
| CGL | 135.27 | 2.75 | 31 |
| GLE-MM | 132.52 | 130e-04 | 124 |
| GLE-ADMM | 132.52 | 1.28e-04 | 49 |

### D. Real-Data Experiments — Genetic Regulatory Networks

We tested GLE-ADMM and GLE-MM on real data from gene expression networks using the two data sets from [22], [36], respectively. (1) Lymph node status and (2) Arabidopsis thaliana. See [36] and references therein for the descriptions of these data sets. Lymph node status is an important clinical risk factor affecting the long-term outlook for breast cancer treatment outcome, and the data consists of 4514 genes from 148 samples. A gene network of Arabidopsis thaliana, which consists of 835 genes monitored using 118 GeneChip (Affymetrix) microarrays, is also studied. The experimental implementation of GLE-ADMM and GLE-MM follows the discussion in Section VI-A and objective value is calculated using (9). The test results are presented in Table IV, and the V. We can see from the tables that although the GLE-ADMM and GLE-MM methods take more CPU time, they consistently outperform the CGL method in terms of the optimality gap. The results reiterate that the fact the CGL solution is not optimal, while the proposed methods are optimal.

## VII. CONCLUSION

In this paper, we have studied the graph Laplacian estimation problem under a given connectivity topology. We have proposed two estimation algorithms, namely, GLE-ADMM and GLE-MM, to improve the optimality performance of the traditional CGL algorithm. Both algorithms can achieve an optimality gap as low as $10^{-4}$, around three orders of magnitude more accurate than the benchmark CGL. In addition, we have found that GLE-ADMM is more efficient in a dense topology, while GLE-MM is more suitable for sparse graphs. Moreover, we have considered exploiting the leading eigenvectors of the sample covariance matrix as a nominal eigensubspace. The simulation results have suggested an improvement in the graph Laplacian estimation when the sample size is smaller than or comparable to the problem dimension. We have proposed a third algorithm, named GLENE, based on ADMM for the inclusion of a nominal eigensubspace. The optimality gap with respect to the CVX toolbox is less than $10^{-4}$. In a real-data experiment, we have shown that GLENE is able to reveal the strong correlations between stocks and, meanwhile, achieve a high sparsity ratio.

## APPENDIX A
### PROOF OF LEMMA 2

*Proof:* The proof is as follows:

$$\arg \min_{\mathbf{C} \in \mathcal{C}} -\text{Tr}\left(\mathbf{Y}^T \mathbf{C}\right) + \frac{\rho}{2} \|\mathbf{X} - \mathbf{C}\|_F^2$$

$$= \arg \min_{\mathbf{C} \in \mathcal{C}} \sum_i \sum_j \left[ -Y_{ij} C_{ij} + \frac{\rho}{2} \left(X_{ij} - C_{ij}\right)^2 \right]$$

$$= \begin{cases} \left[\frac{1}{\rho} Y_{ij} + X_{ij}\right]_+ & i = j \\ 0 & i \neq j, A_{ij} = 0 \\ \left[\frac{1}{\rho} Y_{ij} + X_{ij}\right]_- & i \neq j, A_{ij} = 1 \end{cases} \quad (59)$$

$$= \mathbf{I} \odot \left[\frac{1}{\rho} \mathbf{Y} + \mathbf{X}\right]_+ + \mathbf{A} \odot \left[\frac{1}{\rho} \mathbf{Y} + \mathbf{X}\right]_- .$$

∎

## REFERENCES

[1] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, Sep. 2017.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, Apr. 2013.

[3] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[4] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*. Boca Raton. FL, USA: CRC Press, 2005.

[5] C. Zhang and D. Florêncio, "Analyzing the optimality of predictive transform coding using graph-based models," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 106–109, Jan. 2013.

[6] C. Zhang, D. Florêncio, and P. A. Chou, "Graph signal processing–A probabilistic framework," Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2015-31, 2015.

[7] 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.

[8] E. Levitan and G. T. Herman, "A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography," *IEEE Trans. Med. Imag.*, vol. MI-6, no. 3, pp. 185–192, Sep. 1987.

[9] S. Krishnamachari and R. Chellappa, "Multiresolution Gauss-Markov random field models for texture segmentation," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 251–267, Feb. 1997.

[10] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 4, pp. 959–963, Aug. 1985.

[11] F. S. Cohen, Z. Fan, and M. A. Patel, "Classification of rotated and scaled textured images using Gaussian Markov random field models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 2, pp. 192–202, Feb. 1991.

[12] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[13] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *J. Mach. Learn. Res.*, vol. 9, no. Mar, pp. 485–516, 2008.

[14] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[15] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graphs," in *Proc. 32nd Annu. Meeting Cogn. Sci. Soc.*, 2010, pp. 778–784.

[16] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. 19th Int. Conf. Artif. Intell. Statist.*, 2016, pp. 920–929.

[17] E. Pavez, H. E. Egilmez, and A. Ortega, "Learning graphs with monotone topology properties and multiple connected components," *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2399–2413, May 2018.

[18] K.-S. Lu and A. Ortega, "Closed form solutions of combinatorial graph Laplacian estimation under acyclic topology constraints," 2017, arXiv:1711.00213.

[19] F. R. Chung, *Spectral Graph Theory*. Providence, RI USA: Amer. Math. Soc., 1997, no. 92.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[21] A. Benfenati, E. Chouzenoux, and J.-C. Pesquet, "A nonconvex variational approach for robust graphical lasso," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 3969–3973.

[22] K. Scheinberg, S. Ma, and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2101–2109.

[23] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[24] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[25] M. W. Jacobson and J. A. Fessler, "An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2411–2422, Oct. 2007.

[26] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

[27] R. Varadhan and C. Roland, "Simple and globally convergent methods for accelerating the convergence of any EM algorithm," *Scand. J. Statist.*, vol. 35, no. 2, pp. 335–353, 2008.

[28] L. Zhao, J. Song, P. Babu, and D. P. Palomar, "A unified framework for low autocorrelation sequence design via majorization–minimization," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 438–453, Jan. 2017.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[30] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.

[31] R. Sun, Z.-Q. Luo, and Y. Ye, "On the expected convergence of randomly permuted ADMM," 2015, arXiv:1503.06387.

[32] Q. Liu, X. Shen, and Y. Gu, "Linearized ADMM for non-convex nonsmooth optimization with convergence analysis," *IEEE Access*, vol. 7, pp. 76131–76144, 2019.

[33] C. Chen, M. Li, X. Liu, and Y. Ye, "On the convergence of multi-block alternating direction method of multipliers and block coordinate descent method," 2015, arXiv:1508.00193.

[34] "The MOSEK optimization toolbox for MATLAB manual, version 7.1 (revision 28)," MOSEK, Tech. Rep., 2015. [Online]. Available: http://www.mosek.com

[35] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[36] L. Li and K.-C. Toh, "An inexact interior point method for l 1-regularized sparse covariance selection," *Math. Program. Comput.*, vol. 2, no. 3/4, pp. 291–315, 2010.

**Licheng Zhao** received the B.S. degree in information engineering from Southeast University, Nanjing, China, and the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2014 and 2018, respectively. His research interests are in optimization theory and fast algorithms, with applications in signal processing, machine learning, and financial engineering.

**Yiwei Wang** received the B.S. degree in information engineering from Southeast University, Nanjing, China, in 2017. She is currently working toward the master's degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Hong Kong. His research interests are in optimization theory, with applications in graphs, machine learning.

**Sandeep Kumar** received the B.Tech. degree from the College of Engineering Roorkee, Roorkee, India, in 2007, and the M.Tech. and Ph.D. degrees from the Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, India, in 2013 and 2017, respectively. He is currently working as a Postdoctoral Researcher with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong. The overarching theme of his research is on algorithms, analysis, and applications of optimization, statistics, and signal processing for data science application.

**Daniel P. Palomar** (S'99–M'03–SM'08–F'12) received the degree in electrical engineering and the Ph.D. degree from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1998 and 2003, respectively, and was a Fulbright Scholar at Princeton University during 2004–2006.

He is currently a Professor with the Department of Electronic and Computer Engineering and the Department of Industrial Engineering & Decision Analytics, Hong Kong University of Science and Technology (HKUST), Hong Kong, which he joined in 2006. He had previously held several research appointments, namely at King's College London, London, U.K.; Stanford University, Stanford, CA, USA; Telecommunications Technological Center of Catalonia, Barcelona, Spain; Royal Institute of Technology (KTH), Stockholm, Sweden; University of Rome "La Sapienza", Rome, Italy; and Princeton University, Princeton, NJ, USA. His current research interests include applications of optimization theory and signal processing in financial systems and big data analytics.

Dr. Palomar is a recipient of a 2004/06 Fulbright Research Fellowship, the 2004 and 2015 (co-author) Young Author Best Paper Awards by the IEEE Signal Processing Society, the 2015–2016 HKUST Excellence Research Award, the 2002/03 best Ph.D. prize in Information Technologies and Communications by the UPC, the 2002/03 Rosina Ribalta first prize for the Best Doctoral Thesis in Information Technologies and Communications by the Epson Foundation, and the 2004 prize for the best Doctoral Thesis in Advanced Mobile Communications by the Vodafone Foundation and COIT.

He has been a Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING 2016 Special Issue on "Financial Signal Processing and Machine Learning for Electronic Trading," an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and IEEE TRANSACTIONS ON SIGNAL PROCESSING, a Guest Editor of the IEEE SIGNAL PROCESSING MAGAZINE 2010 Special Issue on "Convex Optimization for Signal Processing," the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2008 Special Issue on "Game Theory in Communication Systems," and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2007 Special Issue on "Optimization of MIMO Transceivers for Realistic Communication Networks".