GRACGE: Graph Signal Clustering and Multiple Graph Estimation

Yanli Yuan, De Wen Soh, Xiao Yang, Kun Guo, *Member, IEEE*, and Tony Q. S. Quek, *Fellow, IEEE*

***Abstract*—In graph signal processing (GSP), complex datasets arise from several underlying graphs and in the presence of heterogeneity. Graph learning from heterogeneous graph signals often results in challenging high-dimensional multiple graph estimation problems, and prior information regarding which graph the data was observed is typically unknown. To address the above challenges, we develop a novel framework called GRACGE (*GRAph signal Clustering and multiple Graph Estimation*) to partition the graph signals into clusters and jointly learn the multiple underlying graphs for each of the clusters. GRACGE advocates a regularized EM (rEM) algorithm where a structure fusion penalty with adaptive regularization parameters is imposed on the M-step. Such a penalty can exploit the structural similarities among graphs to overcome the curse of dimensionality. Moreover, we provide a non-asymptotic bound on the estimation error of the GRACGE algorithm, which establishes its computational and statistical guarantees. Furthermore, this theoretical analysis motivates us to adaptively re-weight the regularization parameters. With the adaptive regularization scheme, the final estimates of GRACGE will geometrically converge to the true parameters within statistical precision. Finally, experimental results on both synthetic and real data demonstrate the performance of the proposed GRACGE algorithm.**

***Index Terms*—Multiple graph learning, graph signal processing (GSP), signal clustering, regularized EM algorithm, nonasymptotic statistical analysis.**

I. Introduction

GRAPHS are mathematical tools used in various fields to represent and analyse complicated data with irregular structures [1]. Particularly, in graph signal processing (GSP) [2], [3], graphs provide effective ways to model large-scale structured data from a signal processing perspective. Applications of such models include graph signal filtering [4], graph signal recovery [5], [6], and graph signal sampling [7], [8]. All the aforementioned graph-based modellings require the graph topology, which is inherent to the dataset, to be known beforehand. However, there are often settings where the graph is not readily available. In these cases, it is essential to learn the underlying graph topology from data, hence permitting subsequent data analysis and processing.

Most existing works learn the underlying graph topology by defining meaningful signal models such that the learned graph can best fit the intrinsic characteristics of observed signals [9]–[12]. For example, in [9], the signal is assumed to follow a degenerate multivariate Gaussian distribution with the graph Laplacian being the precision matrix. Under this assumption, the learned graph captures well the smoothness of the signal. Beyond the smooth graph signal model [9], [10], the authors in [11] and [12] focus on learning the graph Laplacian from a more general family of graph signals result from graph diffusion processes, which are defined as exponential functions of the underlying graph Laplacian. These aforementioned models are only suitable for simple datasets, assuming that all samples were identically distributed and correspond to a single graph. However, datasets in many application areas usually come in more complex forms and include heterogeneous samples from several distinct subpopulations. For example, gene expression measurements are often collected from both healthy subjects and patients diagnosed with different subtypes of cancer [13]. User profiles are usually observed from social networks where the same set of users can have different types of social interactions [14]. These heterogeneous samples naturally form clusters, where signals from each of the clusters live on a different graph. Therefore, the structures of such complex datasets require multiple graph Laplacians to model, giving rise to the need for learning multiple graphs.

Multiple graph learning from heterogeneous graph signals is challenging. If we employ a single graph learning method for each cluster separately, we would lose inference benefits that come from the similarities shared among different clusters [15]. We would also require every cluster to have adequate amounts of data samples, which may not always be the case due to cost or time restraints [16]. To overcome these obstacles, it has been shown that joint estimation of multiple graphs can help improve the efficiency of inference by exploiting common statistics of data between different clusters [16]–[18]. Nevertheless, to do

Manuscript received March 10, 2021; revised November 1, 2021 and February 14, 2022; accepted April 8, 2022. Date of publication April 13, 2022; date of current version May 2, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ali Tajer. This work was supported in part by the National Research Foundation, Singapore through its AI Singapore Programme under Grant AISG-RP-2019-015, in part by the SUTD Growth Plan Grant for AI, and in part by Shanghai Pujiang Program under Grant 21PJ1402600. *(Corresponding author: Kun Guo.)*

Yanli Yuan, De Wen Soh, and Tony Q. S. Quek are with the Singapore University of Technology and Design, Singapore (e-mail: yanli_yuan@sutd.edu.sg; dewen_soh@sutd.edu.sg; tonyquek@sutd.edu.sg).

Xiao Yang is with the State Key Laboratory of Integrated Service Institute of Information Science, Xidian University, Xi'an, Shaanxi 710071, China (e-mail: xyang_2@stu.xidian.edu.cn).

Kun Guo is with the Shanghai Key Laboratory of Multidimensional Information Processing, School of Communications and Electronics Engineering, East China Normal University, Shanghai 200241, China (e-mail: guokun1218@foxmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2022.3167145, provided by the authors.

Digital Object Identifier 10.1109/TSP.2022.3167145

so, the above works require that prior knowledge of cluster memberships is available, i.e., they needs to know in advance which samples come from which graphs. This may be a strict requirement in some settings. For instance, in social networks [14], it is not clear which types of social interactions between users are available due to privacy restrictions, causing the need to separate data samples corresponding to different graphs. Hence, we treat the data labels as latent variables to be learned alongside the multiple graph Laplacians.

In this paper, we precisely consider the problem of learning multiple graphs from unlabelled heterogeneous graph signals. To handle this problem, we propose to unify the **GRA**ph signal **C**lustering and multiple **G**raph **E**stimation into one optimization framework, called **GRACGE**. In particular, GRACGE adopts a high-dimensional regularized Expectation-Maximization (rEM) algorithm [20], which is able to group the graph signals into clusters and jointly learn multiple graph Laplacians, one for each of the clusters. In each iteration of the rEM algorithm, the E-step performs graph signals clustering by estimating missing labels and the M-step conducts joint estimation of multiple graph Laplacians via a regularization procedure. With an iteratively updating process, the estimations for both data labels and graph Laplacians become increasingly refined. Specifically, the M-step solves a regularized multiple graph estimation problem, where a structured fusion penalty is introduced to overcome the curse of dimensionality by borrowing strength over the entire dataset [19]. With the data labels found in the E-step, the optimization problem in the M-step becomes convex, which can then be solved efficiently. For example, the ADMM-based algorithm from our previous work can be used to do so [19].

Our main contributions are a set of theoretical results that allow us to establish computational and statistical guarantees for the proposed GRACGE algorithm.

First, from the theory of regularized M-estimators (e.g., [21]), in order to achieve a certain accurate solution, the regularization parameter should be chosen proportional to the target estimation error. In the GRACGE algorithm, the regularized multiple graph estimation problem (M-step) is solved iteratively and the estimation error changes at each iteration, which motivates us to design an adaptive regularization scheme. Hence, we directly evaluate the estimation error in each iteration of the GRACGE algorithm. Such a result provides a theoretical guideline for constructing a regularization sequence so that it converges to a quantity controlled by the final estimation error. With this regularization scheme, in each iteration, GRACGE is performed with a regularization parameter that is adaptive to its previous output, bringing the next output closer to the true parameter.

Second, we provide a precise characterization of the computational complexity and statistical accuracy of the proposed GRACGE algorithm. Doing so requires specialized technical analysis of the relationship between the population and empirical M-step operators in GRACGE. We validate sufficient conditions satisfied by the M-step operators to ensure convergence of the GRACGE algorithm. We show that the estimates of GRACGE will geometrically coverage to the true parameters within a statistical accuracy. This theoretical analysis provides explicit bounds on the sample complexity and computational complexity required by the convergence of GRACGE, which illustrates the impact of the sample size, the number of graphs, and the structural similarity among graphs on the convergence rate and the final statistical accuracy. We further conduct extensive simulation experiments to justify such theoretical analysis.

This paper is organized as follows. We first describe some related works in Section II, and then provide the preliminaries and problem formulation in Section III. Section IV presents the proposed GRACGE algorithm. Section V establishes theoretical guarantees for GRACGE. Experimental results and conclusions are presented in Sections VI and VII, respectively. The proofs of theorems and lemmas are provided in Appendix and Supplements.

II. Related Work

In this section, we will discuss related work. Learning a single graph Laplacian from a signal representation perspective is a well-studied topic [10], [11], [22], and two overview papers of these GSP-based graph learning methods have recently been published [23], [24]. On the basis of single graph learning techniques, many methods have been developed to jointly learn multiple graph Laplacians from signals that naturally live on different graphs. For example, Yamada *et al.* [16] proposed to learn time-varying graphs from spatiotemporal measurements. Subsequently, Segarra *et al.* [25] proposed to jointly infer the topology of a collection of networks from stationary signals. Presently, in our previous work [19], we have developed a general framework for joint estimation of multiple graph Laplacians, which can capture various topological properties among graphs through a structured fusion regularization. However, all aforementioned works crucially rely on an assumption that the cluster label of each sample is known in advance, differing from our current context of unknown data labels.

In order to learn graphs from unlabelled heterogeneous graph signals, a K-graphs method [26] was proposed for learning mixtures of graph Laplacians, which supposes that graph signals are independent and smooth on different graphs. In [27], a dynamic graph learning algorithm, called dynamic K-graphs was further developed that considers the time dependency of signals. Note that both methods in [26] and [27] are based on a smooth signal model, and cannot capture different spectral characteristics of graphs signals. Among the techniques for learning mixtures of graphs, an algorithm called graph Laplacian mixture model (GLMM) [28], which adopts graph-based filters (functions of graph Laplacians) to model various classes of graph signals with specific frequency characteristics, is close to ours. GLMM first decouples the signals into clusters and then applies existing single graph learning methods for each cluster separately. Due to the separate estimation, GLMM cannot overcome the curse of dimensionality suffered by the high-dimensional graph learning problem. Unlike GLMM, GRACGE takes advantage of the previous method [19] using a structured fusion penalty to regularize the entries in the graph Laplacians, which can exploit the common statistics of the entire dataset to improve statistical estimation efficiency. Moreover, in these aforementioned works, the focus is on problem formulation and algorithmic

design, without providing theoretical guarantees. In this paper, a non-asymptotic bound on estimation error of the GRACGE algorithm is provided, which enables us to investigate the effects of several key factors on the computational complexity and statistical accuracy of the GRACGE algorithm.

Other than GSP-based graph learning methods, there have been approaches in developing nonparametric graphical model selection (GMS) for time series by focusing on the frequency domain [29]–[31]. However, these methods assume that the entire set of signals can be represented well on a single graphical model. On the other hand, there are also works aimed at learning multiple parameters of graphical models. For example, several methods for multiple graph estimation are based on Gaussian mixture models (GMMs) [32]–[34]. Note that these works focus on learning the precision matrices without assuming the Laplacian constraints. As a result, the learned graphs have both positive and negative edge weights, which may be unsuitable for interpreting the structures of data in some contexts, such as the correlations between asset returns in financial markets [35], [36]. In contrast, GRACGE works with the combinatorial graph Laplacian matrix, which can be used to perform harmonic analysis of graph signals and is often desirable in GSP [2], [10].

Another line of related work has to do with convergence analysis of EM algorithms. In this work, GRACGE aims at estimating multiple graph Laplacians via a regularized EM (rEM) algorithm. To the best of our knowledge, there is no prior work on the theoretical understanding of a consistency guarantee of such estimator. Nevertheless, there have been convergence analysis of EM algorithms on learning GMMs [20], [37], [38]. Specifically, the work in [37] first utilised the population and sample-based analysis to provide statistical guarantees for such EM algorithms in low-dimensional setting. The theoretical analysis of these high-dimensional EM algorithms was provided in [38], which required specialized treatment of M-step in different settings. The work in [20] generalized the high-dimensional EM algorithm through a unified treatment using regularization (so-called regularized EM algorithm). Our theoretical proofs are built on tools from [20]. However, our analysis procedure does not match their framework: First, their methods are not directly applicable to the estimation of graph Laplacians due to the assumed identity precision matrix. In fact, our consideration of the graph Laplacian as a general precision matrix demands more challenging technical analysis, and we employ nice properties of the M-estimators framework in [39] to address these technical issues. Second, in our work, connecting the population-level results to sample-based results requires specifically addressing the structured fusion penalty, rather than the ℓ_1 penalty in [20].

Notations: For $K \in \mathbb{N}$, $[K]$ denotes the integer set $\{1, \ldots, K\}$. $I(\cdot)$ denotes as an indicator function. Let $a, \mathbf{a}, \mathbf{A}$, and $\mathcal{A}$ denote a scalar, a vector, a matrix, and a set, respectively. $[a]_+ = \max(a, 0)$ and $[a]_- = \min(a, 0)$. $[\mathbf{A}]_+ = \max(\mathbf{A}, 0)$, $[\mathbf{A}]_- = \min(\mathbf{A}, 0)$, $\mathbf{A} \geq 0$ means $\mathbf{A}$ is entry-wise larger than 0, $\mathbf{A} \succeq 0$ means $\mathbf{A}$ is a positive semi-definite matrix. $\sigma(\mathbf{A})$ represents the singular value of a symmetric matrix $\mathbf{A}$. $\|\cdot\|_p$ denotes the ℓ_p-norm of a vector. We define $\|\cdot\|_{1,\text{off}}$ and $\|\cdot\|_{\max,\text{off}}$ as the ℓ_1-norm and ℓ_∞-norm applied to the off-diagonal matrix entries, respectively. $\nabla(\cdot)$ represents the gradient of a multivariate function. $\mathbf{1}$ stands for the all-one vector, and $\mathbf{I}$ stands for the identity matrix. Let $\mathbf{A}^\top, \mathbf{A}^{-1}, \mathbf{A}^\dagger, \mathrm{tr}(\mathbf{A}), \det(\mathbf{A}), |\mathbf{A}|_+$, and $\mathrm{vec}(\mathbf{A})$ denote the transpose, inverse, pseudo-inverse, trace, determinant, pseudo-determinant, and the vectorized form of $\mathbf{A}$, respectively. The (i, j)-th entry of $\mathbf{A}$ is denoted as A_{ij} or $[\mathbf{A}]_{ij}$, the Frobenius norm of $\mathbf{A}$ is $\|\mathbf{A}\|_\mathrm{F} = \sqrt{\sum_{i,j} \mathrm{A}_{ij}^2}$. The operator $\otimes$ denotes the Kronecker product between matrices. The cardinality of the set $\mathcal{A}$ is represented as $\mathrm{card}\,(\mathcal{A})$. For two functions $f(n)$ and $g(n)$, $f(n) = \mathcal{O}(g(n))$ denotes that f is bounded above by g asymptotically; $f(n) = \Omega(g(n))$ denotes that f is bounded below by g asymptotically; $f(n) = o(g(n))$ denotes that f is dominated by g asymptotically; $f(n) \lesssim g(n)$ denotes that $f(n) \leq Cg(n)$ for some absolute constant $C > 0$.

III. Preliminaries and Problem Formulation

A. GSP-Based Graph Learning Fundamentals

The GSP-based graph learning methods require us to define meaningful models for explaining the relations between the unknown graphs and the observed signals. In this paper, we adopt a filtered graph signal model, where the observed signals are outputs of a system with a graph-based filter given certain input, such that the learned graph can capture desirable frequency characteristics of the observed signals [24].

We represent a weighted and undirected graph by a triple $G = (\mathcal{V}, \mathcal{E}, \mathbf{L})$, where $\mathcal{V}$ is a finite set of p nodes, $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the edge set, and $\mathbf{L} \in \mathbb{R}^{p \times p}$ is the combinatorial graph Laplacian matrix representing the graph topology. Specifically, the absolute value of the (i, j)-th entry of $\mathbf{L}$ is denoted as $|\mathrm{L}_{ij}|$, representing the edge weight between node i and j. In general, the set of valid graph Laplacian matrices is written as

$$\mathcal{L} = \left\{ \mathbf{L} \in \mathbb{R}^{p \times p} \mid \mathbf{L} \succeq 0, \mathbf{L} \cdot \mathbf{1} = 0, \mathrm{L}_{ij} = \mathrm{L}_{ji} \leq 0, i \neq j. \right\}. \tag{1}$$

From a GSP perspective, the Laplacian matrix allows for filtering operations of signals on graphs via spectral decomposition. Formally, let the eigendecomposition of a graph Laplacian be $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{\Lambda}$ is a diagonal matrix with ascending ordered non-negative eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_p$ (graph frequencies), and a larger eigenvalue corresponds to a higher graph frequency. For a given signal $\mathbf{x} = [x_1, x_2, \ldots, x_p]^\top \in \mathbb{R}^p$ defined on a graph with p nodes, where x_i represents the signal value of node i, the graph Fourier transform (GFT) of $\mathbf{x}$ is defined as $\hat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$ and the inverse GFT is given by $\mathbf{U}\hat{\mathbf{x}}$. With the notion of GFT, a graph-based filter can be defined as a smooth transfer function $h(\lambda_i)$ which amplifies or attenuates each of the frequency components of the graph signal. Exploiting the filtering operations on graphs, a family of graph signals can be generated by

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{U}h(\mathbf{\Lambda})\mathbf{U}^\top \mathbf{x}_0 = \boldsymbol{\mu} + h(\mathbf{L})\mathbf{x}_0, \tag{2}$$

where $\boldsymbol{\mu}$ is the mean vector. Generally, the input data $\mathbf{x}_0$ is assumed to be multivariate white Gaussian noise, which ensures the stationarity assumption on the observed signals, thereby leading to the filtered graph signal model [11]:

$$\mathbf{x} = \boldsymbol{\mu} + h(\mathbf{L})\mathbf{x}_0 \sim \mathcal{N}\left(\boldsymbol{\mu}, h(\mathbf{L})\mathbf{I}h(\mathbf{L})^\top\right) = \mathcal{N}\left(\boldsymbol{\mu}, h^2(\mathbf{L})\right). \tag{3}$$

The signal model (3) provides a unified representation of stationary signals on a graph and has the unique advantage of capturing

desirable frequency characteristics of the observed signals. For example, to enforce a priori information that the observed signal $\mathbf{x}$ is smooth with respect to the underlying graph, the graph-based filter has the form $h(\mathbf{L}) = \sqrt{\mathbf{L}^\dagger}$, i.e., $h(\lambda_i) = \begin{cases} 1/\sqrt{\lambda_i} & \lambda_i > 0 \\ 0 & \lambda_i = 0 \end{cases}$, which attenuates high frequency components of graph signals and thus can be viewed as a low-pass filter [9], [10].

Remark 1: As stated in [24], depending on the appropriate graph-based filer, signals generated by the filtered graph signal model mainly fall into two distinct types, i.e., globally smooth signals (e.g., $h(\mathbf{L}) = \sqrt{\mathbf{L}^\dagger}$ studied in [9], [10]) and signals from graph diffusion processes (e.g., $h(\mathbf{L}) = e^{-\nu\mathbf{L}}$ studied in [12]). These two types of graph signals are popular in many applications. We justify that the GRACGE algorithm is adaptable to learning graphs from these two types of observed signals. In this paper, we provide detailed algorithmic steps for learning from smooth signals. For applications where the graph diffusion processes are presented, the GRACGE algorithm and its corresponding theoretical analysis can be applied with some modifications, which will be considered in our future work.

B. Unlabelled Heterogeneous Graph Signals

In this paper, we will deal mainly with unlabelled heterogeneous graph signals from several distinct subpopulations. Suppose the graph signals reside on K related, but distinct graphs $G_1, \ldots, G_K$. Each graph $G_k = \{\mathcal{V}, \mathcal{E}_k, \mathbf{L}_k\}$ has a specific set of edges $\mathcal{E}_k$ that are characterized by the Laplacian matrix $\mathbf{L}_k \in \mathcal{L}$. Given that a graph signal $\mathbf{x} \in \mathbb{R}^p$ is attached to a designated graph G_k, i.e., the corresponding graph signal model is given by

$$f_k(\mathbf{x} \mid \mathbf{L}_k) = \mathcal{N}\left(\boldsymbol{\mu}_k, h^2(\mathbf{L}_k)\right), \tag{4}$$

We consider that signal samples from the same graph naturally form a cluster. In practice, we do not know necessarily which graph a signal sample comes from. Hence, a probabilistic model is needed to accommodate the latent cluster label in the data. Let $\mathbf{Z}$ denote the unknown graph indication matrix whose entry $\{\mathrm{Z}_{ki}; k \in [K], i \in [n]\}$ is a latent variable of whether the i-th sample $\mathbf{x}_i$ is derived from the graph G_k, so $\mathrm{Z}_{ki} = 1$ if $\mathbf{x}_i$ is related to G_k and $\mathrm{Z}_{ki} = 0$ otherwise. Assume a prior distribution on $\mathbf{Z}$ with probability $\Pr(\mathrm{Z}_{ki} = 1) = g_k$, then the unlabelled heterogeneous graph signals $\{\mathbf{x}_i; i \in [n]\}$ can be modelled by a graph Laplacian mixture model [28], which is given by

$$f(\mathbf{x}_i) = \sum_{k=1}^{K} g_k f_k(\mathbf{x}_i \mid \mathbf{L}_k). \tag{5}$$

C. Problem Formulation

The primary problem of this paper is the learning mixtures of graphs from unlabelled heterogeneous signals, which can be formulated as a maximum-likelihood estimation. Formally, given that the total n signal samples form a data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{p \times n}$, the empirical log-likelihood is

$$\mathcal{F}_n(\mathbf{L}; \mathbf{X}) = \frac{1}{n} \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} g_k f_k(\mathbf{x}_i \mid \mathbf{L}_k) \right). \tag{6}$$

Here, from now on, $\mathbf{L} = \{\{\mathbf{L}_k\}_{k=1}^K \mid \mathbf{L}_k \in \mathcal{L}, k \in [K]\}$ denotes the set of multiple graph Laplacians.

In the high-dimensional setting where $p \gg n$, it is well known that the maximum likelihood estimator is not consistent unless additional constraints are imposed on the model [39]. To overcome the curse of dimensionality, we advocate a maximum penalized likelihood estimation, which is defined as

$$\begin{aligned} \underset{\{g_k, \boldsymbol{\mu}_k, \mathbf{L}_k\}_{k=1}^K}{\text{maximize}} \quad & \mathcal{F}_n(\mathbf{L}; \mathbf{X}) - \rho_n \mathcal{P}(\mathbf{L}) \\ \text{subject to} \quad & \mathbf{L}_k \in \mathcal{L}, k \in [K], \\ & \times \sum_{k=1}^{K} g_k = 1. \end{aligned} \tag{7}$$

Here, $\rho_n > 0$ is a regularization parameter and $\mathcal{P}(\mathbf{L})$ is a convex penalty function, which encourages similarity between different graph $\mathbf{L}_k$. By enforcing structural similarity, each $\mathbf{L}_k$ borrows strength from the fact that related graph estimates should be similar, or even identical, and thus it is able to overcome the issue of some graphs having insufficient samples. In this paper, we adopt a structured fusion penalty introduced in [19], which is given by

$$\begin{aligned} \mathcal{P}(\mathbf{L}) &= \mathcal{P}_1(\mathbf{L}) + \rho \mathcal{P}_2(\mathbf{L}) \\ &= \sum_{i \neq j} \|\mathbf{L}_{ij}\|_1 + \rho \sum_{i \neq j} \sqrt{\mathbf{L}_{ij}^\top \widetilde{\mathbf{J}} \mathbf{L}_{ij}}, \end{aligned} \tag{8}$$

where $\mathbf{L}_{ij} = ([\mathbf{L}_1]_{ij}, \ldots, [\mathbf{L}_K]_{ij})^\top \in \mathbb{R}^K, i, j \in [p]$ is a vector of (i,j)-entries across the K graph Laplacians, $\widetilde{\mathbf{J}} = \mathbf{J}^\top \mathbf{J}$ is a Gram matrix with $\mathbf{J}$ being a given matrix, and $\rho > 0$ is a tuning parameter. The penalty in (8) combines a ℓ_1-norm with a weighted ℓ_2-norm. The ℓ_1 norm $\mathcal{P}_1(\mathbf{L})$ induces individual sparsity of each graph. While the weighted ℓ_2 norm $\mathcal{P}_2(\mathbf{L})$ represents a family of the group-structured norms which has the natural tendency of fusing each group of coefficients according to their correlations [39]. The matrix $\widetilde{\mathbf{J}}$ typically reflects some underlying topological pattern across K graphs. More generally, the matrix $\mathbf{J}$ can be chosen so that the learned graphs exhibit desired structural similarity depending on the context. For an illustration, we set $\mathbf{J} = \mathbf{I}$, then the penalty $\mathcal{P}_2(\mathbf{L}) = \sum_{i \neq j} \sqrt{\sum_{k=1}^K [\mathbf{L}_k]_{ij}^2}$ corresponds to the group graphical Lasso [40], which encourages a similar pattern of sparsity across all graphs, i.e., there will be a tendency for the zeros in all estimated Laplacian matrices $\hat{\mathbf{L}}_k, k \in [K]$ to occur at the same places.

Remark 2: As justified in [19], the structured fusion penalty is flexible enough to allow for the recovery of many different forms of structural similarities among graphs via different choices of $\mathbf{J}$. In practice, if we have prior information about the structural similarity among graphs, we can encode it into $\mathbf{J}$. It is worth mentioning that the GRACGE algorithm and its theoretical analysis framework are usable with any choice of $\mathbf{J}$ without any modification.

IV. The GRACGE Algorithm

In this section, we introduce the GRACGE algorithm, which leverages the high-dimensional rEM algorithm [20] to solve the non-convex optimization problem in (7). The general scheme of the rEM algorithm is to apply the classic EM algorithm with an adaptive regularization sequence. Hence, as summarized in Algorithm 1, GRACGE involves three main steps, which are repeated until convergence is achieved. In the sequel, we will elaborate on the involved three main steps in GRACGE.

1) Graph Signal Clustering (E-Step): Here, we perform a soft clustering procedure on data, which computes the posterior probability of the cluster label for each data. Let $\mathbf{\Gamma}$ denote the conditional probability of $\mathbf{Z}$ given $\mathbf{X}$, whose entry $\Gamma_k(\mathbf{x}_i;\mathbf{L}) = \Pr(\mathrm{Z}_{ki}=1 \mid \mathbf{x}_i)$. For simplicity, we will use the shorthand Γ_{ki} for $\Gamma_k(\mathbf{x}_i;\mathbf{L})$. The value of Γ_{ki} can be computed using Bayes' theorem

$$\begin{aligned}\Gamma_{ki} &= \frac{\Pr(\mathrm{Z}_{ki}=1)\Pr(\mathbf{x}_i \mid \mathrm{Z}_{ki}=1)}{\sum_{j=1}^{K}\Pr(\mathrm{Z}_{ji}=1)\Pr(\mathbf{x}_i \mid \mathrm{Z}_{ji}=1)} \\ &= \frac{g_k f_k(\mathbf{x}_i \mid \mathbf{L}_k)}{\sum_{j=1}^{K} g_j f_j(\mathbf{x}_i \mid \mathbf{L}_j)}. \end{aligned} \tag{9}$$

We shall view Γ_{ki} as the *responsibility* that graph G_k takes for 'explaining' the observation $\mathbf{x}_i$. Thus, we can compute data labels based on $\mathbf{\Gamma}$.

2) Multiple Graph Estimation (M-Step): Once $\mathbf{\Gamma}$ is obtained, the log-likelihood $\mathcal{F}_n(\mathbf{L};\mathbf{X},\mathbf{\Gamma})$ can be lower bounded by

$$\begin{aligned}&\mathcal{F}_n(\mathbf{L};\mathbf{X},\mathbf{\Gamma}) \\ &= \frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\Gamma_{ki}\frac{g_k f_k(\mathbf{x}_i \mid \mathbf{L}_k)}{\Gamma_{ki}}\right) \\ &\overset{(a)}{\geq} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\left(\Gamma_{ki}\log g_k f_k(\mathbf{x}_i \mid \mathbf{L}_k) - \Gamma_{ki}\log\Gamma_{ki}\right), \end{aligned} \tag{10}$$

where (a) follows from Jensen's inequality. We denote the first term in (10) as the surrogate $\mathcal{Q}$-function, i.e.,

$$\mathcal{Q}_n(\mathbf{L};\mathbf{X},\mathbf{\Gamma}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\Gamma_{ki}\log g_k f_k(\mathbf{x}_i \mid \mathbf{L}_k). \tag{11}$$

sGiven that Γ_{ki} has been obtained from (9), therefore, performing maximization of $\mathcal{F}_n(\mathbf{L};\mathbf{X},\mathbf{\Gamma})$ is equal to maximize its lower bound $\mathcal{Q}$-function. With $\mathcal{Q}_n(\mathbf{L};\mathbf{X},\mathbf{\Gamma})$ at hand, the original formulation (7) is equivalent to

$$\begin{aligned}\underset{\{g_k,\boldsymbol{\mu}_k,\mathbf{L}_k\}_{k=1}^K}{\text{maximize}} \quad & \mathcal{Q}_n(\mathbf{L};\mathbf{X},\mathbf{\Gamma}) - \rho_n\mathcal{P}(\mathbf{L}) \\ \text{subject to} \quad & \mathbf{L}_k \in \mathcal{L}, k\in[K], \\ & \sum_{k=1}^{K} g_k = 1. \end{aligned} \tag{12}$$

As we can see, (12) is a differentiable convex optimization problem with respect to g_k and $\boldsymbol{\mu}_k$, which admits a closed-form solution for both:

$$g_k = \sum_{i=1}^{n}\frac{\Gamma_{ki}}{n}, \boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n}\Gamma_{ki}\mathbf{x}_i}{\sum_{i=1}^{n}\Gamma_{ki}}. \tag{13}$$

To maximize over $\mathbf{L}$, we only need to consider the terms related to $\mathbf{L}$ in (12) and solve the following optimization problem

$$\begin{aligned}\underset{\{\mathbf{L}_k\}_{k=1}^K}{\text{maximize}} \quad & \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\Gamma_{ki}\log f_k(\mathbf{x}_i \mid \mathbf{L}_k) - \rho_n\mathcal{P}(\mathbf{L}) \\ \text{subject to} \quad & \mathbf{L}_k \in \mathcal{L}, k\in[K]. \end{aligned} \tag{14}$$

The optimization problem (14) represents a generic formulation for jointly learning of multiple graph Laplacians. In order to provide the algorithm with more specific and detailed steps, we consider in the rest of the section the special case of the low-pass filter, i.e., $h(\mathbf{L}_k) = \sqrt{\mathbf{L}_k^{\dagger}}$. The low-pass filter results in smooth observed signals, which was noticeably studied in previous graph learning works [9], [10], [41]. Substituting the explicit form of the penalty $\mathcal{P}(\mathbf{L})$ specified in (8), the special case of the generic multiple graph estimation problem (14) is given by

$$\begin{aligned}\underset{\{\mathbf{L}_k\in\mathcal{L}\}_{k=1}^K}{\text{maximize}} & \frac{1}{n}\sum_{k=1}^{K} n_k\left[\log(|\mathbf{L}_k|_+) - \mathrm{tr}(\mathbf{\Sigma}_k\mathbf{L}_k)\right] \\ & \rho_n\left(\sum_{i\neq j}\|\mathbf{L}_{ij}\|_1 + \rho\sum_{i\neq j}\sqrt{\mathbf{L}_{ij}^{\top}\widetilde{\mathbf{J}}\mathbf{L}_{ij}}\right), \end{aligned} \tag{15}$$

where $\mathbf{\Sigma}_k = \frac{\sum_{i=1}^{n_k}\Gamma_{ki}(\mathbf{x}_i-\boldsymbol{\mu}_k)(\mathbf{x}_i-\boldsymbol{\mu}_k)^{\top}}{\sum_{i=1}^{n}\Gamma_{ki}}$ and $n_k = \sum_{i=1}^{n}\Gamma_{ki}$. The problem (15) is a convex problem, where the penalty term

Algorithm 1: Graph Signal Clustering and Multiple Graph Estimation (GRACGE).

Input: Observed signals $\{\mathbf{x}_i\}_{i=1}^n$, the number of graphs K, number of iterations T, initial parameters $\{g_k^{(0)}=1/K\}_{k=1}^K$, $\{\boldsymbol{\mu}_k^{(0)}\}_{k=1}^K$, $\mathbf{L}^{(0)}=\{\mathbf{L}_k^{(0)}\}_{k=1}^K$, initial regularization parameter $\rho_n^{(0)}$, structural similarity matrix $\mathbf{J}$, quantity ε, the tuning parameter $\rho>0$, contraction factor $\kappa\in(0,1)$.

Output: Cluster label matrix $\mathbf{\Gamma}$, graph Laplacians $\{\mathbf{L}_k\}_{k=1}^K$;

Begin:

1: **for** iteration $t=1$ to T **do**
2: **Adaptive regularization:** Update $\rho_n^{(t)}$ via (16).
3: **E-step (Graph signal clustering) :**Calculate the cluster label $\Gamma_{ki}^{(t)}$according to (9) for each $k\in[K], i\in[n]$.
4: **M-step (Multiple graph estimation):**
1. Update $g_k^{(t)}, \boldsymbol{\mu}_k^{(t)}$ based on (13) for each $k\in[K]$.
2. Call JEMGL algorithm in [19] to solve (15) for the update of $\mathbf{L}^{(t)}$.
5: **end for**
6: **return** $\mathbf{\Gamma}^{(T)}, \mathbf{L}^{(T)}$.

$\sum_{i\neq j}\sqrt{\mathbf{L}_{ij}^{\top}\widetilde{\mathbf{J}}\mathbf{L}_{ij}}$ makes the parameters in $\mathbf{L}=\{\mathbf{L}_k\}_{k=1}^K$ coupled together. Therefore, we use an ADMM-based algorithm called JEMGL, which was originally developed in [19], to solve the problem (15). Following the ADMM framework, JEMGL introduces consensus variables to decouple the fused parameters, thus splitting the problem (15) into a series of sub-problems. Moreover, JEMGL has derived closed-form solutions for each of the sub-problems, allowing (15) to be solved efficiently. More details about the JEMGL algorithm can be found in [19].

3) Adaptive Regularization: Problem (14) or its special case (15) can be seen as a regularized M-estimator [39]. In order to ensure the desired convergence of a M-estimator, an appropriate choice of the regularization parameter ρ_n should be chosen proportional to its estimation error [21]. In the GRACGE algorithm, (15) is solved iteratively, motivating the need to adaptively re-weight the regularization parameter ρ_n based on the estimation error in each iteration. As we will show later in Theorem 1, the estimation error in each iteration consists of two terms: an optimization error term that decays geometrically with the number of iteration and a statistical error term that does not depend on the iteration number, but rather on the observed data of the problem. Hence, we iteratively update the regularization parameter via a recursion of the form

$$\rho_n^{(t)}=\kappa\rho_n^{(t-1)}+\varepsilon. \tag{16}$$

Here, t represents the iteration index, ε characterizes the target statistical error, and $\kappa\in(0,1)$ is a contraction factor.

The regularization sequence $\rho_n^{(t)}$ stated in (16) plays an important role in controlling the final estimation error of the GRACGE algorithm. In the following Section V, we establish the theoretical guarantees for the GRACGE algorithm and provide the precise update for $\rho_n^{(t)}$ in Corollary 1.

V. Theoretical Guarantees

In this section, we will describe the theoretical analysis of the GRACGE algorithm. We derive a non-asymptotic bound of the estimation error in each iteration of GRACGE. To formally state our analysis, we first describe some notations. Next, we prove some key ingredients that are necessary conditions on the population and empirical M-step operators in GRACGE. Finally, we present our main results and their interpretations.

A. Notations

To treat multiple graph Laplacians in a unified matrix form, let $\Omega\subset\mathbb{R}^{pK\times pK}$ be the space of symmetric K-block diagonal matrices, where each diagonal block corresponds to a graph Laplacian. Let $\mathbf{L}^*\in\Omega$ be the true Laplacian matrices of K graphs. At the t-th iteration, given the previous output $\mathbf{L}^{(t-1)}\in\Omega$, GRACGE computes the surrogate $\mathcal{Q}$-function from a finite number n of samples according to (11), which is given by

$$\mathcal{Q}_n\left(\mathbf{L}'\mid\mathbf{L}^{(t-1)}\right)=\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\Gamma_k(\mathbf{x}_i;\mathbf{L}^{(t-1)})\log g_k f_k(\mathbf{x}_i\mid\mathbf{L}_k'). \tag{17}$$

We name $\mathcal{Q}_n(\cdot|\cdot)$ as the sample-based $\mathcal{Q}$-function and the population-level counterpart $\mathcal{Q}(\cdot|\cdot)$ corresponds to

$$\mathcal{Q}\left(\mathbf{L}'\mid\mathbf{L}^{(t-1)}\right)=\mathbb{E}_X\left[\sum_{k=1}^{K}\Gamma_k(X;\mathbf{L}^{(t-1)})\log g_k f_k(X\mid\mathbf{L}_k')\right], \tag{18}$$

where $\mathbf{L},\mathbf{L}'\in\Omega$, X is a random variable with the marginal distribution defined in (5), and $\Gamma_k(X;\mathbf{L})=\mathbb{E}[\Gamma_k(\mathbf{x}_i;\mathbf{L})]$. In order to compactly represent the update of the M-step in GRACGE, we define the empirical operator $\mathcal{M}_n:\Omega\to\Omega$ as

$$\mathcal{M}_n(\mathbf{L}):=\underset{\mathbf{L}'\in\Omega}{\arg\max}\,\mathcal{Q}_n\left(\mathbf{L}'\mid\mathbf{L}\right). \tag{19}$$

The penalized empirical operator $\mathcal{M}_n^p:\Omega\to\Omega$ is defined as

$$\mathcal{M}_n^p(\mathbf{L}):=\underset{\mathbf{L}'\in\Omega}{\arg\max}\,\mathcal{Q}_n(\mathbf{L}'\mid\mathbf{L})-\rho_n\mathcal{P}(\mathbf{L}'). \tag{20}$$

The population operator $\mathcal{M}:\Omega\to\Omega$ is defined as

$$\mathcal{M}(\mathbf{L}):=\underset{\mathbf{L}'\in\Omega}{\arg\max}\,\mathcal{Q}\left(\mathbf{L}'\mid\mathbf{L}\right). \tag{21}$$

Assume the oracle surrogate function $\mathcal{Q}(\cdot|\mathbf{L}^*)$ is self consistent, namely

$$\mathbf{L}^*\equiv\mathcal{M}(\mathbf{L}^*). \tag{22}$$

The output of the GRACGE algorithm at the t-th iteration corresponds to $\mathbf{L}^{(t)}=\mathcal{M}_n^p(\mathbf{L}^{(t-1)})$.

Our goal is to provide an upper bound of the estimator error at each iteration in the Frobenius norm, i.e., $\|\mathbf{L}^{(t)}-\mathbf{L}^*\|_{\mathrm{F}}=\|\mathcal{M}_n^p(\mathbf{L}^{(t-1)})-\mathcal{M}(\mathbf{L}^*)\|_{\mathrm{F}}$. Note that this error bound depends on the relationship between the population and penalized empirical M-step operators. In the following section V-B, we set up a general analytical framework for GRACGE where the key ingredients are decomposable penalty $\mathcal{P}(\cdot)$ and several technical conditions about population-level $\mathcal{Q}(\cdot|\cdot)$ and sample-based $\mathcal{Q}_n(\cdot|\cdot)$. We validate these conditions in Lemma 1-4.

B. Key Ingredients

1) Decomposable Penalty: Here, we prove the decomposable properties satisfied by the structured fusion penalty (8) in Lemma 1. Since we are interested in bounding the estimation error in the Frobenius norm $\|\cdot\|_{\mathrm{F}}$, the goal here is to connect the penalty function $\mathcal{P}(\cdot)$ to the norm $\|\cdot\|_{\mathrm{F}}$.

Denote the support space of $\mathbf{L}_k^*$ as $\mathcal{S}_k=\{(i,j):[\mathbf{L}_k]_{ij}\neq 0,i,j=1,\ldots,p,i\neq j\}$, then the support space of $\mathbf{L}^*$ is $\mathcal{S}=\bigcup_{k=1}^K\mathcal{S}_k$. The orthogonal complement of support space $\mathcal{S}$, namely, is defined as the set

$$\mathcal{S}^{\perp}:=\{\mathbf{L}'\in\Omega\mid\langle\mathbf{L},\mathbf{L}'\rangle=0,\forall\mathbf{L}\in\mathcal{S}\}. \tag{23}$$

Given a matrix set $\mathbf{L}\in\Omega$, we use $\mathbf{L}_{\mathcal{S}}$ to denote the projection of $\mathbf{L}$ onto $\mathcal{S}$.

Lemma 1: (i) Our penalty defined in (8) is a seminorm, convex, and decomposable with respect to $(\mathcal{S},\mathcal{S}^{\perp})$, i.e.,

$$\mathcal{P}(\mathbf{L}_1+\mathbf{L}_2)=\mathcal{P}(\mathbf{L}_1)+\mathcal{P}(\mathbf{L}_2),\forall\mathbf{L}_1\in\mathcal{S},\mathbf{L}_2\in\mathcal{S}^{\perp}. \tag{24}$$

Besides,

$$\mathcal{P}(\mathbf{L}^*+\mathbf{\Delta})-\mathcal{P}(\mathbf{L}^*)\geq\mathcal{P}\left(\mathbf{\Delta}_{\mathcal{S}^\perp}\right)-\mathcal{P}\left(\mathbf{\Delta}_{\mathcal{S}}\right). \tag{25}$$

(ii) The dual norm of $\mathcal{P}(\mathbf{L})$ represented by $\mathcal{P}^*(\mathbf{L})$ can be bounded by

$$\mathcal{P}^*(\mathbf{L})\leq\left(1+\sigma_{\min}(\widetilde{\mathbf{J}})\sqrt{K}\right)\max_k\|\mathbf{L}_k\|_{\max,\text{off}}. \tag{26}$$

(iii) For $\mathbf{L}\in\mathcal{S}$, $\mathcal{P}(\mathbf{L})$ satisfies the following inequality:

$$\mathcal{P}(\mathbf{L})\leq\Psi(\mathcal{S})\cdot\|\mathbf{L}\|_{\mathrm{F}}, \tag{27}$$

where $\Psi(\mathcal{S})\leq\sqrt{s}(1+\rho\sqrt{\sigma_{\max}(\widetilde{\mathbf{J}})})$ and s represents the sparsity parameter, i.e., $s:=\text{card}(\mathcal{S})$.

Proof: Lemma 1 is the same as the Lemma 1 in [19]. Please see [19] for proof. □

Remark 3: We justify that the technical tools used in the proof of Lemma 1 are also adaptable to other convex and decomposable penalties. Therefore, our analytical framework for the GRACGE algorithm is usable with a wide array of convex and decomposable penalties.

2) Population-Level Analysis: As given in (18), even with an infinite number of samples, the performance of optimizing the population-level function $\mathcal{Q}(\cdot|\cdot)$ (i.e., the objective function of the multiple graph estimation problem) depends on the performance of signal clustering (i.e., the accuracy of the estimated cluster label matrix $\mathbf{\Gamma}$). This additional difficulty of simultaneous clustering and estimation can be characterized by the following sufficiently separable condition. Define a ball $\mathbb{B}_r(\mathbf{L}^*):=\{\mathbf{L}\in\Omega\mid\|\mathbf{L}-\mathbf{L}^*\|_{\mathrm{F}}\leq r\}$.

Assumption 1: (*Separable Condition*) Denote $I=\max_{j\in[K]}I_j, I'=\max_{j\in[K]}I'_j$, with I_j, I'_j defined in (76). Let $H=\sqrt{\max\{I,I'\}}$. Assume that K clusters are sufficiently separable such that given an appropriately small parameter $\tau>0$, it hold a.s.

$$\Gamma_k(X,\mathbf{L})\cdot\Gamma_j(X,\mathbf{L})\leq\frac{\tau}{2H(K-1)}, \tag{28}$$

for each pair $\{(k,j),k,j\in[K],k\neq j\}$ and any $\mathbf{L}\in\mathbb{B}_r(\mathbf{L}^*)$.

The product $\Gamma_k(X,\mathbf{L})\cdot\Gamma_j(X,\mathbf{L})$ represents a degree of separation between the densities of the k-th and j-th clusters in the mixture. When the two clusters are well-separated, $\Gamma_k(X,\mathbf{L})\cdot\Gamma_j(X,\mathbf{L})$ becomes 0, otherwise $\Gamma_k(X,\mathbf{L})\cdot\Gamma_j(X,\mathbf{L})$ certainly takes a higher value. Assumption 1 requires that the distributions of K clusters are sufficient separable in the sense that the signal X belongs to the k-th cluster with probability either 0 or close to 1. This is consistent with the main theorem in [42]. As stated in (28), the separable condition is related with the number of clusters K, as K grows, the problem gets harder and hence a stronger condition is needed.

In (21), the population operator relies on the quantity $\nabla\mathcal{Q}(\cdot|\mathbf{L})$, and the global optimum in (22) depends on the value of the quantity $\nabla\mathcal{Q}(\cdot|\mathbf{L}^*)$. Thus, we are naturally led to consider a gradient smoothness condition that ensures the closeness of these two quantities. The next Lemma 2 guarantees the curvature of $\mathcal{Q}(\cdot|\mathbf{L})$ is similar to that of $\mathcal{Q}(\cdot|\mathbf{L}^*)$ when $\mathbf{L}$ is close to $\mathbf{L}^*$.

Lemma 2: (Gradient Stability) Under the Assumption 1 and for any $\mathbf{L}\in\mathbb{B}_r(\mathbf{L}^*)$, the population-level function $\mathcal{Q}(\cdot|\cdot)$ satisfies

$$\|\nabla\mathcal{Q}\left(\mathbf{L}^\diamond\mid\mathbf{L}\right)-\nabla\mathcal{Q}\left(\mathbf{L}^\diamond\mid\mathbf{L}^*\right)\|_{\mathrm{F}}\leq\tau\cdot\|\mathbf{L}-\mathbf{L}^*\|_{\mathrm{F}}, \tag{29}$$

where $\mathbf{L}^\diamond=\mathcal{M}(\mathbf{L})$ defined in (21).

Proof: Please refer to Supplements-A. □

3) Sample-Based Analysis: Similar strong concavity condition is widely used in convex optimization and plays an important role in showing geometric convergence of gradient descent. Here, to guarantee the estimation error $\|\mathbf{L}^{(t)}-\mathbf{L}^*\|_{\mathrm{F}}$ in the t-th iteration of the GRACGE algorithm is well controlled, we expect that the sample-based $\mathcal{Q}_n(\cdot\mid\mathbf{L}^{(t-1)})$ is strongly concave at $\mathbf{L}^*$. To verifies the restricted strong concavity condition of $\mathcal{Q}_n(\cdot\mid\mathbf{L}^{(t-1)})$, we need the following assumptions hold.

Assumption 2: (Regularity Condition) There exist a positive constant c_1 such that $0<\max_{k\in[K]}(\sigma_{\max}(\mathbf{L}_k^*))<c_1$.

Assumption 3: The number of data samples scales as $n=\Omega(\log p)$.

With the above assumptions in hand, we validate the restricted strong concavity of $\mathcal{Q}_n(\cdot\mid\mathbf{L}^{(t-1)})$ in Lemma 3.

Lemma 3: (Restricted Strong Concavity) Suppose that Assumption 2 holds, and with sufficient large n satisfying Assumption 3, there is a strong concavity parameter $\gamma=\frac{c_2}{(c_1+r)^2}$, where $c_2>0$ is a bounded constant defined in (89). Then for any fixed $\mathbf{L}\in\mathbb{B}_r(\mathbf{L}^*)$ and each $\mathbf{L}'\in\mathbb{B}_r(\mathbf{L}^*)$, we have that

$$\begin{aligned}&\mathcal{Q}_n\left(\mathbf{L}'|\mathbf{L}\right)-\mathcal{Q}_n\left(\mathbf{L}^*|\mathbf{L}\right)-\left\langle\nabla\mathcal{Q}_n\left(\mathbf{L}^*|\mathbf{L}\right),\mathbf{L}'-\mathbf{L}^*\right\rangle\\&\quad\leq-\frac{\gamma}{2}\|\mathbf{L}'-\mathbf{L}^*\|_{\mathrm{F}}^2\end{aligned} \tag{30}$$

with probability at least $1-\beta_1$.

Proof: Please refer to Supplements-B. □

4) From Population to Sample-Based Analysis: In the GRACGE algorithm, the M-step computes the penalized empirical operator in (20) from a finite number of samples, which is an approximation of the population operator defined in (21). Due to finite samples, such an approximation leads to the statistical error. In Lemma 4, we provide an upper bound of the statistical error.

Assumption 4: The largest element of the covariance matrices are bounded, that is, there exists a positive constant c_6 such that $\max_{k\in[K]}\|(\mathbf{L}_k^*)^\dagger\|_{\max,\text{off}}\leq c_6$.

Lemma 4: (Statistical Error) Suppose there exists some absolute constant $C>c_6$. For any fixed $\mathbf{L}\in\mathbb{B}_r(\mathbf{L}^*)$, with probability at least $1-\beta_2$, we have

$$\mathcal{P}^*\left(\nabla\mathcal{Q}_n\left(\mathbf{L}^*|\mathbf{L}\right)-\nabla\mathcal{Q}\left(\mathbf{L}^*|\mathbf{L}\right)\right)\leq\varepsilon_n, \tag{31}$$

where $\mathcal{P}^*(\cdot)$ is the dual norm of $\mathcal{P}(\cdot)$, $\beta_2=K^2\beta$, and

$$\varepsilon_n=C\left(1+\sigma_{\min}(\widetilde{\mathbf{J}})\sqrt{K}\right)K\sqrt{\frac{\log p+\log(e/\beta)}{n}}. \tag{32}$$

Proof: Please refer to Supplements-C. □

The parameter ε_n characterizes the achievable statistical error. Note that $\lim_{n\to\infty}\varepsilon_n=0$, which suggests that we can obtain zero statistical error with an infinite number of samples. As we will see in the sequel, the final estimation error of the GRACGE

algorithm is the same order of the statistical error, hence, ε_n plays an important role in determining the radius of convergence.

Remark 4: Based on the result in (32), we observe that the structural similarity among graphs quantified by the matrix $\widetilde{\mathbf{J}}$ plays a role in determining the value of ε_n. More specifically, the term $(1+\sigma_{\min}(\widetilde{\mathbf{J}})\sqrt{K})$ comes from the upper bound of the dual norm $\mathcal{P}^*(\mathbf{L})$ in (26). While an explicit characterization of the spectral properties of the Gram matrix is difficult, investigating some special cases can provide insight into the effect of the underlying structural similarities on the statistical error. Consider, for instance, two extreme cases: for a separate ℓ_2 penalty on each graph, then the ℓ_2 norm in (8) becomes $\mathcal{P}_2(\mathbf{L})=\sum_{k=1}^{K}\sqrt{\sum_{i\neq j}[\mathbf{L}_k]_{ij}^2}$. In this case, the dual norm is bounded by

$$\mathcal{P}^*(\mathbf{L}) \leq (1+\sqrt{p})\max_k \|\mathbf{L}_k\|_{\max,\text{off}}. \tag{33}$$

For the group graphical lasso case, i.e., $\widetilde{\mathbf{J}}=\mathbf{I}$ and $\sigma_{\min}(\widetilde{\mathbf{J}})=1$, we have the dual norm

$$\mathcal{P}^*(\mathbf{L}) \leq \left(1+\sqrt{K}\right)\max_k \|\mathbf{L}_k\|_{\max,\text{off}}. \tag{34}$$

Comparing the two bounds in (33) and (34), we can see that the joint estimation of multiple similar graphs can get a much smaller statistical error than the separate estimation. The above results match the intuition that accounting for structural similarities can improve statistical estimation efficiency through borrowing information across the multiple graphs.

C. Main Results

With the above notations and ingredients in place, we state our main result on the convergence of the GRACGE algorithm.

Theorem 1: Consider the graph Laplacian mixture model (5) with any fixed $\mathbf{L}^*\in\Omega$ and a graph-based filter taking the form $h(\mathbf{L}_k)=\sqrt{\mathbf{L}_k^{\dagger}}$. Suppose Assumptions 1, 2, 3, 4 hold, and the initialization $\mathbf{L}^{(0)}$ lies in a ball $\mathbb{B}_r(\mathbf{L}^*)\subset\Omega$ for some constant radius $r>0$. If the choice of regularization parameter satisfies

$$\rho_n^{(t)} \geq \varepsilon_n + \frac{\tau}{\Psi(\mathcal{S})}\left\|\mathbf{L}^{(t-1)}-\mathbf{L}^*\right\|_{\text{F}}, \tag{35}$$

and when the sample size n is large enough such that $\varepsilon_n \leq (1-\kappa)\gamma r/(5\Psi(\mathcal{S}))$, then with probability at least $1-t\beta'$, we have

$$\left\|\mathbf{L}^{(t)}-\mathbf{L}^*\right\|_{\text{F}} \leq \underbrace{\kappa^t\left\|\mathbf{L}^{(0)}-\mathbf{L}^*\right\|_{\text{F}}}_{(A)} + \underbrace{\frac{5\Psi(\mathcal{S})}{(1-\kappa)\gamma}\varepsilon_n}_{(B)}, \tag{36}$$

where $\kappa=\frac{5\tau}{\gamma}\in(0,1)$ with $\Psi(\mathcal{S}),\tau,\gamma,\varepsilon_n$ are calculated in Lemma 1, Lemma 2, Lemma 3, and Lemma 4, as well as $\beta'=\beta_1+\beta_2$ with β_1 and β_2 defined in Lemma 3 and Lemma 4.

Proof: Theorem 1 is proved in Appendix-A1. □

Theorem 1 establishes computational and statistical guarantees of the GRACGE algorithm. The result in (36) suggests that with suitable regularization parameters, the estimation error is bounded by two terms. The term (A) resulting from iterative optimization of function $\mathcal{Q}_n$ thus is referred to as *optimization error*. The term (B) called *statistical error* characterizes the final estimation error of the GRACGE algorithm. In particular, the parameter $\kappa\in(0,1)$ is contractive and hence the optimization error decays geometrically with the number of iterations t, while the statistical error is independent of t. Such results show that if an appropriate initialization is given, the estimates of the GRACGE algorithm will converge to a local optima or stationary point but within a statistical accuracy of the global optima.

The bound (36) measures the estimation error of the GRACGE algorithm in each iteration, which enables us to provide a meaningful choice of the maximum number of iterations T. When $t\geq T$, the optimization error will be dominated by the statistical error, implying that the whole estimation error bound is in the same order of the statistical error. In the following corollary, we provide a closed form of T and an explicit form of the final estimation error.

Assumption 5: The number of clusters K satisfies $K=o(\log p)$.

Corollary 1: Suppose Assumptions 1, 2, 3, 4, and 5 hold. With a proper choice of β' such as $\beta_1=\beta_2=\frac{1}{p}$, then we can perform the adaptive regularization scheme (16) with the initialization parameters being specified as follows

$$\rho_n^{(0)} = \frac{\gamma}{5\sqrt{s}\left(1+\rho\sqrt{\sigma_{\max}(\widetilde{\mathbf{J}})}\right)}\left\|\mathbf{L}^{(0)}-\mathbf{L}^*\right\|_{\text{F}},$$

$$\varepsilon = \mathcal{O}\left(\left(K+\sigma_{\min}(\widetilde{\mathbf{J}})K^{1.5}\right)\sqrt{\frac{\log p}{n}}\right). \tag{37}$$

If sample size n is sufficiently large such that

$$n \geq \left(\frac{5C\left(1+\rho\sqrt{\sigma_{\max}(\widetilde{\mathbf{J}})}\right)\left(K+\sigma_{\min}(\widetilde{\mathbf{J}})K^{1.5}\right)}{(1-\kappa)\gamma r}\right)^2 s\log p, \tag{38}$$

and the iteration number is given by

$$T \geq \log_{1/\kappa}\frac{\left\|\mathbf{L}^{(0)}-\mathbf{L}^*\right\|_{\text{F}}}{\Delta(n,p,K,s)}, \tag{39}$$

we have the final estimation error as

$$\left\|\mathbf{L}^{(T)}-\mathbf{L}^*\right\|_{\text{F}} \leq 2\Delta(n,p,K,s), \tag{40}$$

with probability converging to 1. Here,

$$\Delta(n,p,K,s) = \frac{\widetilde{C}\left(1+\rho\sqrt{\sigma_{\max}(\widetilde{\mathbf{J}})}\right)\left(K+\sigma_{\min}(\widetilde{\mathbf{J}})K^{1.5}\right)}{(1-\kappa)\gamma}\sqrt{\frac{s\log p}{n}} \tag{41}$$

represents the upper bound of the obtainable statistical error for some positive constant $\widetilde{C}$.

Proof: Corollary 1 is proved in Appendix-A2. □

The results in (37) provide a precise theoretical guideline for updating the regularization parameters (16) in Algorithm 1. In practice, the accurate estimation of the initialization error

$\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_{\mathrm{F}}$ is not required. Based on the first term in (36), overestimating initialization error will potentially increase the total number of iterations but has no essential impact on the final estimation error.

The bound (38) characterizes the sample size required by the GRACGE algorithm for consistency estimation. Except for the signal dimension p, the sparsity parameter s, and the number of clusters K, we note that the strong concavity parameter γ and the spectral properties of the Gram matrix $\widetilde{\mathbf{J}}$ have an impact on the sample complexity. On one hand, the strong concavity parameter γ has the same order as the parameter τ in (29), ensuring the contraction factor $\kappa = \frac{5\tau}{\gamma} \in (0, 1)$. As stated in Assumption 1, the parameter τ reflects the difficulty of separating the clusters. When K is fixed, the condition (28) requires a smaller τ for a better separation of two clusters, while a smaller τ (i.e., a smaller γ in (38)) indicates a higher sample complexity. On the other hand, the structural similarity among graphs represented by the Gram matrix $\widetilde{\mathbf{J}}$ mainly affects the achievable statistical error ε_n in (32), please see discussions in remark 4. Typically, the structured fusion penalty can improve statistical estimation efficiency through borrowing information from related graphs. Hence, fewer samples are needed to consistently estimate multiple graphs that share greater similarities.

The bound (39) explicitly addresses the trade-off between the statistical accuracy and computation complexity of the GRACGE algorithm.

In summary, the results stated in Corollary 1 can not only help us choose the initialization parameters for running the GRACGE algorithm, but also give theoretical guideline in terminating the GRACGE iterations.

VI. Experiments

In this section, we discuss an order selection procedure and demonstrate the superior numerical performance of our method. We first introduce the general experimental settings and then report the experimental results.

A. Experimental Settings

1) Baselines for Comparison: We compare the GRACGE algorithm with two related signal clustering and graph learning algorithms: GLMM [28] and a two-stage method (denoted as k-means+JEMGL) which first uses a k-means algorithm [43] to obtain the data labels and then applies the JEMGL algorithm [19] for jointly estimating multiple graph Laplacians.

2) Evaluation Metrics: The Clustering Error (CE) metric can measure the normalized mean squared error (NMSE) of signal cluster identification [28], which is computed by

$$\mathrm{CE} = \frac{1}{2n}\|\mathbf{\Gamma} - \mathbf{Z}\|_{\mathrm{F}}^2, \tag{42}$$

where $\mathbf{\Gamma}, \mathbf{Z}$ are the estimated and ground-truth cluster identification matrix. The F-score (FS) metric can measure the accuracy of graph topology recovery, which is given by

$$\mathrm{FS} := \frac{1}{K}\sum_{k=1}^{K}\frac{2\mathrm{tp}_k}{2\mathrm{tp}_k + \mathrm{fn}_k + \mathrm{fp}_k}. \tag{43}$$

Where $\mathrm{tp}_k, \mathrm{fn}_k$, and fp_k represent *true positive*, *false negative*, and *false positive*, respectively. The FS, which is the harmonic average of the precision and recall, reflects the quality of the estimated graph topology. The FS takes values between 0 and 1. The higher the FS is, the better the performance of capturing graph topology is.

3) Synthetic Data Generation: In the simulation, we create K random graphs that share a similar topology. Let the graph Laplacian matrix be $\mathbf{L}_k = \mathbf{L}_c + \mathbf{U}_k, k \in [K]$, where $\mathbf{L}_c$ is shared in all graphs and $\mathbf{U}_k$ represents the unique structure of the k-th graph. The common part $\mathbf{L}_c$ is the Laplacian matrix of a 30-node (i.e., $p = 30$) undirected graph, which follows an Erdos-Renyi model [44] with an edge connection probability 0.2. For each $\mathbf{U}_k$, we first set $\mathbf{U}_k = \mathbf{0}$, and then we randomly pick m pairs of symmetric off-diagonal zero entries and replace them with value -1. Here, we define $\eta = 1 - \frac{m}{M}$ to represent the topological similarity among K graphs with M denoting the total number of edges in each graph. Finally, we modify the diagonal entries in each $\mathbf{L}_k$ to ensure it's a valid Laplacian matrix (i.e., $\mathbf{L}_k \in \mathcal{L}$). After obtaining the graph Laplacian $\mathbf{L}_k$, we randomly generate a signal $\mathbf{x}_i$ through a distribution $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{L}_k^{\dagger})$, where the means for each cluster are randomly drawn from $\boldsymbol{\mu}_k \sim \mathcal{N}(0, 25\mathbf{I})$. In the following experiments, the dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ consists of n samples, and each sample in the dataset comes with equal probability (i.e., $1/K$) from the K graphs.

4) Order Selection: In the signal clustering and graph estimation formulation, the number of graphs K need to be appropriately selected when it is unknown beforehand. Since all the graph learning algorithms are the maximum likelihood-type estimator, we can adopt a BIC-type selection criterion, which admits high data likelihood while preferring small model complexity. Formally, for a fixed K, the BIC criterion is defined as

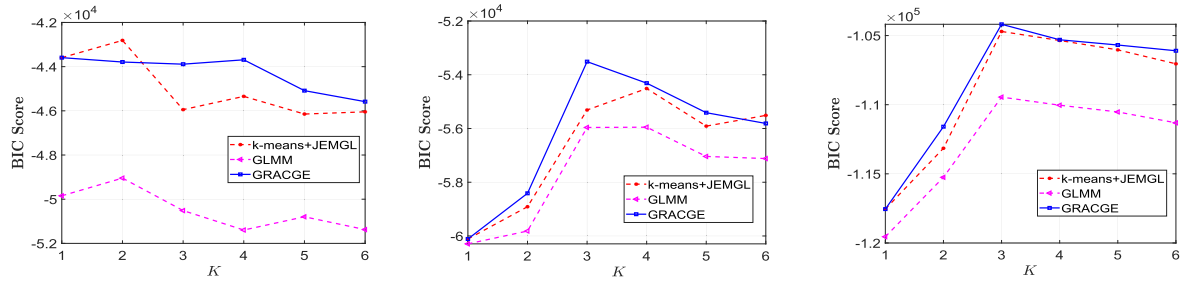
$$\mathrm{BIC}(K) = \widehat{\mathrm{PLL}}(K) - \mathrm{df}(K)\log n, \tag{44}$$

where $\widehat{\mathrm{PLL}}(K)$ is the penalized sample log-likelihood function, $\mathrm{df}(K)$ is the degree of freedom (i.e., model complexity) of a K-graph mixture model, and n is the sample size used for estimating the parameters. According to the graph Laplacian mixture model assumption, the BIC criterion in (44) can be computed as

$$\begin{aligned}\mathrm{BIC}(K) = &\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\widehat{g}_k f_k\left(\mathbf{x}_i; \widehat{\boldsymbol{\mu}}_k, \widehat{\mathbf{L}}_k\right)\right)\\ &- \mathcal{P}(\widehat{\mathbf{L}}^{(K)}) - (K - 1 + K(p + p(p+1)/2)\log n.\end{aligned} \tag{45}$$

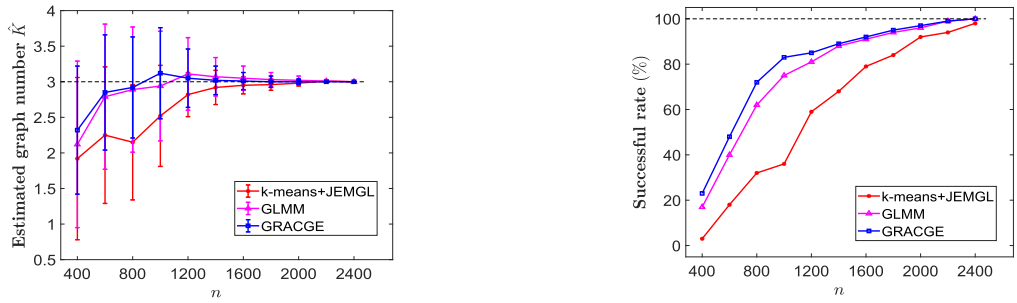
where $\widehat{g}_k, \widehat{\boldsymbol{\mu}}_k, \widehat{\mathbf{L}}_k$ are the final updates of a graph learning algorithm. In the experiment, we choose the optimal graph number maximizing the BIC score in (45).

5) Implementation: For all algorithms, the initial parameter $\mathbf{L}^{(0)}$ is picked from a ball $\{\mathbf{L} : \|\mathbf{L} - \mathbf{L}^*\|_{\mathrm{F}} = 0.5\|\mathbf{L}^*\|_{\mathrm{F}}\}$ uniformly at random and all algorithms are terminated when the Frobenius norm of the change of $\mathbf{L}$ between iterations is smaller than a threshold (by default 10^{-3}). The Gram matrix in both GRACGE and JEMGL is set to $\widetilde{\mathbf{J}} = \mathbf{I}$. In GRACGE, the choice of $\rho_n^{(0)}$ and ε are given in (37). We chose the tuning parameter



(a) sample size $n = 500$ (b) sample size $n = 1000$ (c) sample size $n = 10000$

Fig. 1. The BIC scores of the three graph learning algorithms versus the number of graphs when (a) sample size is 500; (b) sample size is 1000; (c) sample size is 10000.



(a) (b)

Fig. 2. The accuracy of order selection via BIC criteria for the three graph learning algorithms over 100 replications: (a) the average estimated number of graphs ($\pm$ std) versus sample size; (b) successful rate (%) in choosing the correct number versus sample size. The black dotted line in (a) signals the true number of graphs.

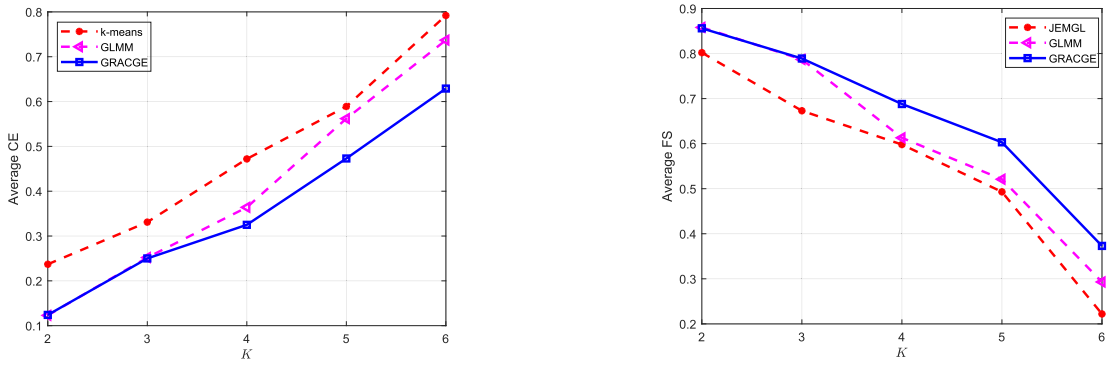
ρ by conducting a grid search over tuning range $10^{-2+2r/15}$ with $r = 0, 1, \ldots, 20$. To tackle the initialization issue, in each setting, we choose the best run in terms of log-likelihood out of 50 runs with different k-means starts.

B. Experiments on Synthetic Data

1) Order Selection Performance: In this experiments, we consider a $K = 3$ graph Laplacian mixture model. We run the tree graph learning algorithms with each of candidate value K over the set $\{1, 2, 3, 4, 5, 6\}$. We first generate date with sample size $n = 500, 1000$ and 10000, and investigate the effect of sample size on the BIC score. Fig. 1(a), (b), and (c) show the BIC scores of the three graph learning algorithms with different sample sizes. It can be observed from Fig. 1 that the BIC scores of all algorithms fluctuate more at smaller sample sizes. Specifically, all the three algorithms perform poorly when $n = 500$. Whereas when $n = 10000$, the BIC scores of all three algorithms have a distinct single peak at $K = 3$, indicating that the number of graphs is correctly identified.

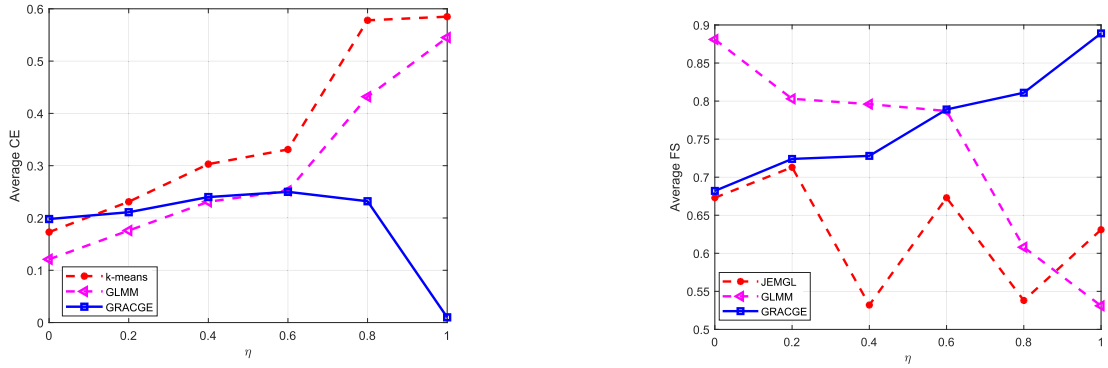
We further investigate the order selection accuracy of all algorithms as the sample size n varies. To access sampling variation, for each n we generate 100 training data sets. To measure the accuracy, we use the averaged estimated graph number ($\pm$ standard deviation) and the successful rate in choosing the correct number (i.e., $\widehat{K} = 3$) over 100 repetitions. The results are visualized in Fig. 2. The results in Fig. 2(a) show that the accuracy of order selection increases with the sample size for all three algorithms. When n is small, e.g., $n < 1400$, none of the algorithms are robust in choosing the true number. On the contrary, when n is large, e.g., $n \geq 2200$, all the algorithms choose the true number with high confidence. These observations match the sample complexity requirement in Assumption 1 for satisfying the *separable condition* on signal clustering. The results in Fig. 2(b) show that both the GRACGE and GLMM algorithms choose the true number with a reasonably high probability when the sample size is at least moderately large, e.g., $n \geq 2200$. However, in the small sample size regime, e.g., $n < 1400$, our GRACGE algorithm gives the best performance. The better performance of GRACGE over GLMM is likely due to the structure fusion penalty adopted in the former. Compared with GLMM, which estimates each graph separately and ignores the similarity across different graphs, GRACGE is able to take advantage of additional sources of information from related graphs and thus enjoys a higher statistical estimation efficiency in high-dimensional settings. On the other hand, compared with the EM-type methods (i.e., GRACGE and GLMM), the two stage method (k-means+JEMGL) reports the lowest successful rate in choosing the correct number in all cases. The two-stage method can be viewed as an EM algorithm with only one iteration, and its performance depends heavily on the initialization value. The above results demonstrate our simultaneous clustering and estimation strategy as well as the consideration of structural similarity across graphs can help improve the order selection performance.

2) Clustering and Graph Estimation Performance: In this subsection, we demonstrate the importance of simultaneous clustering and estimation in improving both the clustering performance and the graph estimation accuracy. To back up our theory, we compare the performance of our GRACGE



(a) Graph signal clustering performance (b) Multiple graph estimation performance

Fig. 3. Performance comparisons with respect to the number of graphs K.



(a) Graph signal clustering performance (b) Multiple graph estimation performance

Fig. 4. Performance comparisons with regard to the topology similarity among graphs η.

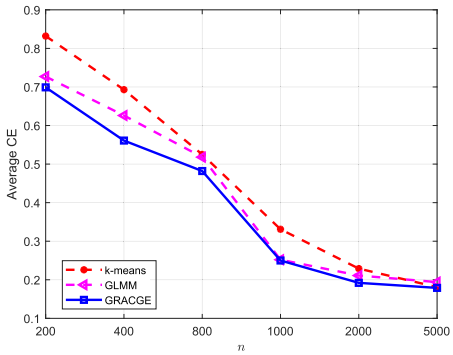
algorithm with the two baseline algorithms under different sets of parameters, i.e., the number of graphs (clusters) K, the level of topological similarity among graphs η, and the sample size n. For a fair comparison, we assume the number of graphs K is given in all methods.

Fig. 3 illustrates how the signal clustering and graph topology inference, measured by CE and FS, change as the number of graphs $K = \{2, 3, 4, 5, 6\}$. In this experiment, we set the topological similarity among graphs is set to $\eta = 0.6$ and generate a dataset $\mathbf{X}$ consisting of $n = 1000$ graph signals. From the results in Fig. 3, it can be seen that the performance of all algorithms in terms of clustering and graph estimation decrease with the growing K. This is because, on the one hand, as K grows, the clustering problem gets harder and more stronger conditions are needed to separate the signals, which is stated in Condition 1. On the other hand, as proved in Corollary 1, the graph estimation precision would also decay with the growing K. In particular, as shown in Fig. 3(a), the worst clustering performance belongs to the k-means method due to its inability to incorporate the graph structure of data into clustering. The benefit of GRACGE over JEMGL can be ascribed to the simultaneously clustering and joint graph estimation via an iterative procedure. Besides, in the two iterative algorithms GLMM and GRACGE, lower clustering error results in better estimation of multiple graphs and vice versa. This suggests that accurate clustering is critical for multiple graph estimation, and alternatively, a good estimation of graphs can substantially improve the clustering performance.

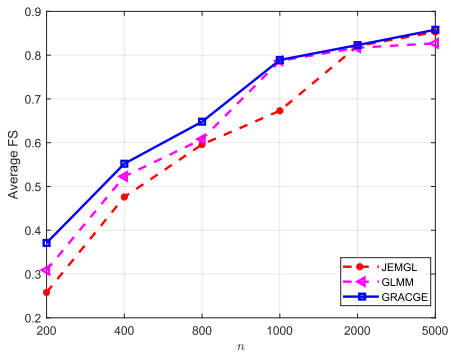
In addition, it can be seen that GLMM and GRACGE show comparable performance in the regime of $K \leq 3$, however, GRACGE performs better than GLMM under larger K. This is because, compared to GLMM, which estimates each graph separately and ignores the similarity across different graphs, the GRACGE can utilize information from related graphs by jointly estimating multiple graphs so as to achieve better performance in high-dimensional settings.

Fig. 4 illustrates the effect of the topological similarity η among graphs on the performance of three algorithms. We conduct experiments on $K = 3$ related graphs with the topological similarity η varying from 0 to 1, keeping the total sample size fixed ($n = 1000$). Note that, GLMM outperforms the other two methods in terms of both clustering and graph topology inference when $\eta \leq 0.4$, indicating that separate estimation is more suitable for estimating multiple independent graphs[1] However, it is also interesting to see that the performance of GLMM becomes worse when $\eta \geq 0.8$, implying that separate estimation might not be optimal for highly related graphs. As for the k-means+JEMGL method, its clustering performance decays with the growing η, while the curves of FS show fluctuations.

[1]Setting the Gram matrix $\widetilde{\mathbf{J}} = \mathbf{0}$, the GRACGE can also perform separate estimation of multiple graphs. Data-adaptive choices of $\widetilde{\mathbf{J}}$ may result in further improvements in the performance of the GRACGE. However, we do not pursue such choices here. We refer the interested readers to [19] for the choice of $\widetilde{\mathbf{J}}$ in detail.



(a) Graph signal clustering performance



(b) Multiple graph estimation performance

Fig. 5. Performance comparisons according to the sample size n.

This is because k-means performs better under small η, whereas JEMGL functions better under large η. Besides, we find that the graph topology inference performance of GRACGE becomes better as η increase. This is consistent with our main Theorem, with the growing η, the topologies of K graphs become more and more similar, and the accuracy of joint graph estimation will improve as a result of exploiting information from each other. These results indicate that as long as there is a substantially similar structure among graphs, joint estimation of multiple graphs is superior to separate estimation.

Fig. 5 illustrates the performance comparisons according to the sample size n. In this experiment, we construct $K = 3$ related graphs with $\eta = 0.6$, and let the sample size n vary from 200 to 5000. In general, the performance of all algorithms increases significantly when more data samples are available, and then changes smoothly when n keeps growing. Because with very small values of n, all algorithms have difficulties in estimating graphs due to the curse of dimensionality, which fit in well with the results stated in Corollary 1. Specifically, the separate estimation algorithm, i.e., GLMM, presents worse performance than the GRACGE algorithm. Note that JEMGL sometimes shows a comparable graph estimation performance with GRACGE when n is large, which indicates that the choice of a graph learning algorithm should make a trade-off between the sample complexity and computation complexity.

In summary, various experiments on synthetic data show that our GRACGE algorithm achieves promising results in comparison with other inspected methods, especially for highly related graphs and in high-dimensional settings.

C. Results on Real Data

We further apply our GRACGE algorithm on real data to demonstrate how this approach can be used to find meaningful insights from heterogeneous graph signals in an unsupervised way.

We analyse the **Webkb** data set[2] from the World Wide Knowledge Base project at Carnegie Mellon University [45].

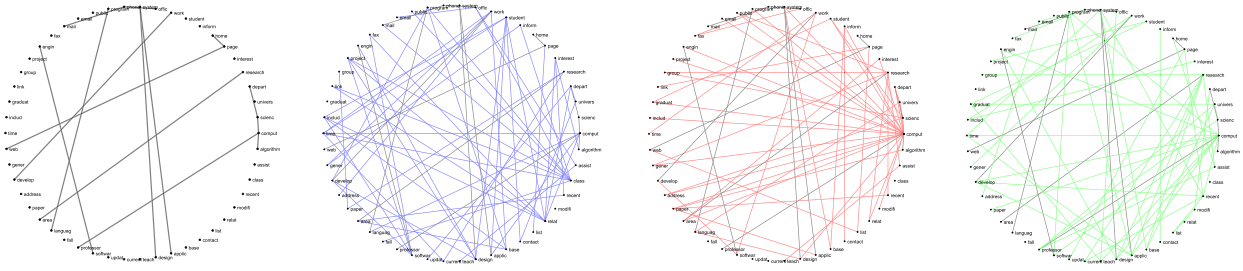
[2]The full data set can be downloaded from the machine learning repository at the University of California, Irvine, http://www. ics.uci.edu/ mlearn/.

TABLE I
CLUSTERING ERROR (CE) OF THE THREE INSPECTED METHODS ON THE **WEBKB** DATA

Methods	k-means	GLMM	GRACGE
CE	0.627	0.436	0.312

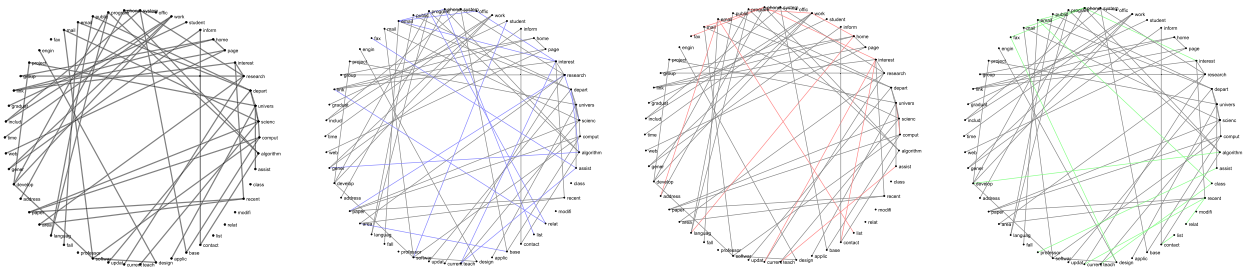
The data set contains webpages from websites at computer science departments in various universities. The webpages are heterogeneous with multiple categories: student, faculty, course, project, staff, department, and the other. The word-word graphs of these different webpage categories are expected to share similarities, but also have unique features. We applied the three graph learning methods to a subset of the **Webkb** data, containing $n = 1228$ webpages from three previously known categories: student (544 webpages), faculty (374 webpages), and course (310 webpages). The original data was preprocessed by Cardoso-Cachopo [46]. The log-entropy weighting method [47] was used to calculate the word-webpage matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$ with p denoting the number of distinct words. In particular, let $f_{ij}, i \in [p], j \in [n]$ be the number of times the i-th word appears in the j-th webpage and let $h_{ij} = \frac{f_{ij}}{\sum_{j=1}^{n} f_{ij}}$. Then, the log-entropy weight of the i-th word is defined as $e_i = 1 + \sum_{j=1}^{n} h_{i,j} \log(h_{i,j})/\log(n)$. Finally, the (i, j)-th entry of the word-webpage matrix is given by $[\mathbf{X}]_{ij} = e_i \log(1 + f_{ij}), i \in [p], j \in [n]$, and it is normalized along each row. For ease of presentation, we focused on $p = 50$ words with the highest log-entropy weights. The goal of the experiment is to classify the 1228 webpages into three categories and construct the word-word network of each category from the word-webpage matrix $\mathbf{X}$.

We first look at clustering performance, which is presented in Table I in terms of clustering error (CE). The standard k-means clustering has the largest clustering error due to its ignorance of incorporating the graph structure in clustering. The GLMM also performed worse than the GRACGE because of estimating each graph individually without borrowing information from each other. Our method is able to achieve the best clustering performance due to the procedure of iteratively between clustering and joint graph learning.



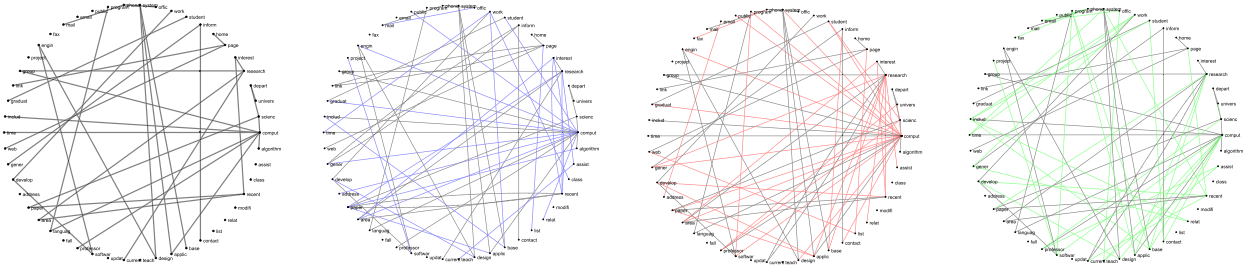
(a) Common Structure (b) 'Course' Graph (c) 'Student' Graph (d) 'Faculty' Graph

Fig. 6. The estimated graphs by the GLMM algorithm.



(a) Common Structure (b) 'Course' Graph (c) 'Student' Graph (d) 'Faculty' Graph

Fig. 7. The estimated graphs by the JEMGL algorithm.



(a) Common Structure (b) 'Course' Graph (c) 'Student' Graph (d) 'Faculty' Graph

Fig. 8. The estimated graphs by the GRACGE algorithm.

We also investigate the graph topology inference performance on this data. As there are no ground-truth graphs for this data, to facilitate the comparison, tuning parameters is selected such that the estimated graphs of the three categories by each method contain about 200 edges in total. Estimated word-word networks of the three categories by GLMM, JEMGL, and GRACGE are shown in Figs. 6, 7, and 8, respectively. In all figures, the black lines are common edges shared in all three categories and the color lines are the unique edges in one or two categories. It can be seen in Fig. 6 that the three estimated graphs by GLMM vary from category to category, and common structure is almost obscured. This variability naturally arise from separate estimation and thus it can't capture similarities among different categories. Compared the graphs in Figs. 7 and 8, we observe that JEMGL finds the more edges common to all three categories, this is probably because the k-means has poor performance of clustering and thus the consequent JEMGL method can't distinguish well between the three categories. Whereas GRACGE can not only capture the common semantic structure of the websites but also allow us to explore the heterogeneity across different categories. For example, in Fig. 6, standard phrases in computer science, such as softwar-develop, program-language, can be found in the common structure. On the other hand, the link assist-professor is only appeared in 'faculty' graph, while research-assist is linked in 'student' graph. The above results highlights the potential advantages of GRACGE in providing graph estimates that can better capture the semantic meaning in websites.

VII. Conclusion

In this paper, we have presented a framework for learning mixtures of graphs from unlabelled heterogeneous graph signals, called GRACGE. GRACGE unifies the graph signal clustering and multiple graph estimation into one optimization framework. Moreover, we provided a non-asymptotic bound on estimation error of the GRACGE algorithm, which gives a clear interpretation of the impact of both the model parameters and the graph structural similarity on the convergence rate and estimation accuracy. We also provided empirical evidence that showed good agreement with our theoretical analysis. Finally, experimental results on real data demonstrated that meaningful insights can be obtained by GRACGE.

Throughout this paper, we have assumed that the number of clusters is known and fixed. In some cases where there is no prior knowledge of the number of clusters, the problem of the empirical estimation of K is of great importance. Although there are some unsupervised EM methods that can automatically determine the optimal number of clusters [48], these methods are computationally expensive. It would be of interest to consider the study of computationally efficient model selection approaches in future research.

APPENDIX A
PROOFS OF MAIN RESULTS

In this section, we provide detailed proofs of main results: Theorem 1 and Corollary 1.

A Proof of Theorem 1

Proof: In order to establish our main theorem, we first present a key Lemma which shows that the choice of regularization parameter $\rho_n^{(t)}$ plays an important role in controlling statistical and optimization error. According to this lemma, we can precisely understand how the statistical error and optimization error accumulate with more and more iterations.

Lemma 5: Let $\mathbf{L}^{(t)} = \mathcal{M}_n^p(\mathbf{L}^{(t-1)})$ be an optimal solution to the optimization problem (20) at the t-th iteration. Suppose $\mathbf{L}^{(t-1)} \in \mathbb{B}_r(\mathbf{L}^*)$, with the choice of a strictly positive regularization parameter satisfying

$$\rho_n^{(t)} \geq \varepsilon_n + \frac{\tau}{\Psi(\mathcal{S})} \left\| \mathbf{L}^{(t-1)} - \mathbf{L}^* \right\|_{\mathrm{F}}, \tag{46}$$

then the estimation error will satisfy

$$\begin{aligned} \left\| \mathbf{L}^{(t)} - \mathbf{L}^* \right\|_{\mathrm{F}} &\leq \frac{5\Psi(\mathcal{S})}{\gamma}\varepsilon_n + \frac{5\tau}{\gamma} \left\| \mathbf{L}^{(t-1)} - \mathbf{L}^* \right\|_{\mathrm{F}} \\ &\leq \frac{5\Psi(\mathcal{S})\rho_n^{(t)}}{\gamma}, \end{aligned} \tag{47}$$

with probability at least $1 - (\beta_1 + \beta_2)$ for all $t = 1, 2, \cdots$. Here $\Psi(\mathcal{S}), \tau, \gamma$ and ε_n are defined in Lemma 1, Lemma 2, Lemma 3 and Lemma 4 accordingly.

The proof of Lemma 5 is postponed to Appendix D.

Equipped with the results of (47) in Lemma 5, we are able to precisely quantify the final estimation error after t iterations. This can be achieved by mathematical induction.

For simplicity, define $\kappa := \frac{5\tau}{\gamma} \in (0, 1)$.

Assume $\mathbf{L}^{(0)} \in \mathbb{B}_r(\mathbf{L}^*)$, then for $t = 1$, applying (47) yields that

$$\left\| \mathbf{L}^{(1)} - \mathbf{L}^* \right\|_{\mathrm{F}} \leq \frac{5\Psi(\mathcal{S})}{\gamma}\varepsilon_n + \kappa \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}}, \tag{48}$$

with probability at least $1 - (\beta_1 + \beta_2)$. Suppose the following inequality is true for some $t \geq 1$,

$$\left\| \mathbf{L}^{(t)} - \mathbf{L}^* \right\|_{\mathrm{F}} \leq \frac{1-\kappa^t}{1-\kappa}\frac{5\Psi(\mathcal{S})}{\gamma}\varepsilon_n + \kappa^t \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}}, \tag{49}$$

with probability at least $1 - t(\beta_1 + \beta_2)$.

Now, We need to verify when $t = t + 1$, the above inequality still holds.

First, we show that $\forall t \geq 0, \mathbf{L}^{(t)} \in \mathbb{B}_r(\mathbf{L}^*)$. Assume the sample size n is large enough, such that

$$\varepsilon_n \leq \frac{(1-\kappa)r\gamma}{5\Psi(\mathcal{S})}. \tag{50}$$

Under the assumption of (50), we have

$$\begin{aligned} \left\| \mathbf{L}^{(t)} - \mathbf{L}^* \right\|_{\mathrm{F}} &\leq \frac{1-\kappa^t}{1-\kappa}\frac{5\Psi(\mathcal{S})}{\gamma}\frac{(1-\kappa)r\gamma}{5\Psi(\mathcal{S})} + \kappa^t \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}} \\ &\leq \left(1 - \kappa^t\right) r + \kappa^t r = r. \end{aligned} \tag{51}$$

Therefore we have $\mathbf{L}^{(t)} \in \mathbb{B}_r(\mathbf{L}^*), \forall t \geq 0$.

Second, substituting the results of (49) in (47), we obtain that

$$\begin{aligned} &\left\| \mathbf{L}^{(t+1)} - \mathbf{L}^* \right\|_{\mathrm{F}} \\ &\leq \frac{5\Psi(\mathcal{S})}{\gamma}\varepsilon_n + \kappa \left\| \mathbf{L}^{(t)} - \mathbf{L}^* \right\|_{\mathrm{F}} \\ &\leq \frac{5\Psi(\mathcal{S})}{\gamma}\varepsilon_n + \kappa \left(\frac{1-\kappa^t}{1-\kappa}\frac{5\Psi(\mathcal{S})}{\gamma}\varepsilon_n + \kappa^t \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}} \right) \\ &= \frac{1-\kappa^{t+1}}{1-\kappa}\frac{5\Psi(\mathcal{S})}{\gamma}\varepsilon_n + \kappa^{t+1} \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}}, \end{aligned} \tag{52}$$

with probability at least $1 - (t+1)(\beta_1 + \beta_2)$.

Consequently, by mathematical induction we can reach the conclusion that

$$\begin{aligned} \left\| \mathbf{L}^{(t)} - \mathbf{L}^* \right\|_{\mathrm{F}} &\leq \frac{1-\kappa^t}{1-\kappa}\frac{5\Psi(\mathcal{S})}{\gamma}\varepsilon_n + \kappa^t \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}} \\ &\leq \frac{5\Psi(\mathcal{S})}{(1-\kappa)\gamma}\varepsilon_n + \kappa^t \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}}, \end{aligned} \tag{53}$$

with probability at least $1 - t(\beta_1 + \beta_2)$.

This completes the proof of Theorem 1.□

B Proof of Corollary 1

Proof: Note that Theorem 1 is true only if condition (46) and condition (50) are met.

First we derive the expression of the initial regularization parameter. Under condition (46), we have

$$\begin{aligned} \rho_n^{(t)} &\geq \varepsilon_n + \frac{\tau}{\Psi(\mathcal{S})} \left\| \mathbf{L}^{(t-1)} - \mathbf{L}^* \right\|_{\mathrm{F}} \\ &\geq \frac{1-\kappa^{t+1}}{1-\kappa}\varepsilon_n + \frac{\tau}{\Psi(\mathcal{S})}\kappa^{t-1} \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}}. \end{aligned} \tag{54}$$

Applying the updating rule of $\rho_n^{(t)}$ in Algorithm 1 implies that

$$\rho_n^{(t)} = \frac{1-\kappa^t}{1-\kappa}\varepsilon + \kappa^t \rho_n^{(0)}. \tag{55}$$

Putting (54) and (55) together and setting $t = 1$ in (54), we can let the initial regularization parameter be

$$\rho_n^{(0)} = \frac{\gamma}{5\sqrt{s}\left(1 + \rho\sqrt{\sigma_{\max}(\widetilde{\mathbf{J}})}\right)} \left\| \mathbf{L}^{(0)} - \mathbf{L}^* \right\|_{\mathrm{F}}. \tag{56}$$

By setting $\beta_2 = K^2\beta = \frac{1}{p}$, we have that

$$\varepsilon \approx \varepsilon_n = C\left(K + \sigma_{\min}(\widetilde{\mathbf{J}})K^{1.5}\right)\sqrt{\frac{\log p}{n}}, \tag{57}$$

as long as the number of clusters K satisfies $K = o(\log p)$.

Second, in order to satisfy the condition (50), the number of sample size n is lower bounded by

$$n \geq \left(\frac{5C\left(1+\rho\sqrt{\sigma_{\max}(\widetilde{\mathbf{J}})}\right)\left(K+\sigma_{\min}(\widetilde{\mathbf{J}})K^{1.5}\right)}{(1-\kappa)\gamma r}\right)^2 s\log p. \tag{58}$$

Third, we provide a stopping rule T. Suppose when $t = T$, the optimization error is equivalent to the statistical error, thus we have

$$\frac{5\Psi(\mathcal{S})}{(1-\kappa)\gamma}\varepsilon_n = \kappa^T\left\|\mathbf{L}^{(0)} - \mathbf{L}^*\right\|_{\mathrm{F}}. \tag{59}$$

For simplicity, let $\Delta(n, p, K, s)$ denotes the statistical error, i.e.,

$$\Delta(n, p, K, s) = \frac{5\Psi(\mathcal{S})}{(1-\kappa)\gamma}\varepsilon_n. \tag{60}$$

Therefore, we have that

$$T = \log_{\frac{1}{\kappa}}\left(\frac{\left\|\mathbf{L}^{(0)} - \mathbf{L}^*\right\|_{\mathrm{F}}}{\Delta(n, p, K, s)}\right). \tag{61}$$

Finally, setting $\beta_1 = \beta_2 = \frac{1}{p}$, we prove that the probability $1 - \frac{2T}{p}$ is close to 1. Plugging in the expression of T in (61), the probability term is bounded by:

$$\begin{aligned}\frac{2T}{p} &\lesssim \frac{\log_{1/\kappa}\left(n/\left(\sqrt{s\log p}\right)\right)}{p}\\ &\lesssim \frac{\log_{1/\kappa} n}{p}.\end{aligned} \tag{62}$$

Under high-dimensional setting, $\frac{2T}{p}$ goes to zero as $p \gg n$.

Putting pieces together, the final estimation error will be upper bounded by

$$\left\|\mathbf{L}^{(T)} - \mathbf{L}^*\right\|_{\mathrm{F}} \leq 2\Delta(n, p, K, s), \tag{63}$$

with probability tending to 1.

Plugging the expressions of $\Psi(\mathcal{S})$ and ε_n into (60), the explicit bound of the statistical accuracy is given by

$$\begin{aligned}&\Delta(n, p, K, s)\\ &\leq \frac{\widetilde{C}\left(1+\rho\sqrt{\sigma_{\max}(\widetilde{\mathbf{J}})}\right)\left(K+\sigma_{\min}(\widetilde{\mathbf{J}})K^{1.5}\right)}{(1-\kappa)\gamma}\sqrt{\frac{s\log p}{n}}\end{aligned} \tag{64}$$

with some positive constant $\widetilde{C}$.

This ends the proof of Corollary 1. □

REFERENCES

[1] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Comput. Surv.*, vol. 40, no. 1, pp. 1–39, Feb. 2008.

[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, May. 2013.

[3] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[4] D. I. Shuman, "Localized spectral graph filter frames: A unifying framework, survey of design considerations, and numerical comparison," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 43–63, Nov. 2020.

[5] A. Jung and N. Tran, "Localized linear regression in networked data," *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp. 1090–1094, Jul. 2019.

[6] A. Jung, A. O. Hero, I. Mara, S. Jahromi, A. Heimowitz, and Y. C. Eldar, "Semi-supervised learning in network-structured data via total variation minimization," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6256–6269, Dec. 2019.

[7] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling on graphs: From theory to applications," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 14–30, Nov. 2020.

[8] Y. Tanaka and Y. C. Eldar, "Generalized sampling on graphs with subspace and smoothness priors," *IEEE. Trans. Signal Process.*, vol. 68, pp. 2272–2286, Mar. 2020.

[9] 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.

[10] 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.

[11] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from filtered signals: Graph system and diffusion kernel identification," *IEEE Trans Signal Inf. Process. Netw.*, vol. 5, no. 2, pp. 360–374, Sep. 2018.

[12] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 3, pp. 481–496, Sep. 2018.

[13] T. C. G. A. R. Network, "Comprehensive genomic characterization de nes human glioblastoma genes and core pathways," *Nature*, vol. 455, pp. 455–1061, 2008.

[14] P. Bindu, P. Thilagam, and D. Ahuja, "Discovering suspicious behavior in multilayer social networks," *Comput. Hum. Behav.*, vol. 73, pp. 568–582, Apr. 2017.

[15] W. Lee and Y. Liu, "Joint estimation of multiple precision matrices with common structures," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1035–1062, 2015.

[16] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning based on sparseness of temporal variation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5411–5415.

[17] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 2826–2830.

[18] S. Segarra, Y. Wang, C. Uhler, and A. G. Marques, "Joint inference of networks from stationary graph signals," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, 2017, pp. 975–979.

[19] Y. Yuan, D. W. Soh, X. Yang, K. Guo, and T. Q. S. Quek, "Joint network topology inference via structured fusion regularization," 2021, *arXiv:2103.03471*.

[20] X. Yi and C. Caramanis, "Regularized EM algorithms: A unified framework and statistical guarantees," in *Proc Adv. Neural Inf. Process. Syst.*, 2015, pp. 1567–1575.

[21] M. J. Wainwright, "Structured regularizers for high-dimensional problems: Statistical and computational issues," *Annu. Rev. Statist. Appl.*, vol. 1, pp. 233–253, 2014.

[22] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, "Learning sparse graphs under smoothness prior," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 6508–6512.

[23] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, May 2019.

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

[25] M. Navarro, Y. Wang, A. G. Marques, C. Uhler, and S. Segarra, "Joint inference of multiple graphs from matrix polynomials," 2020, *arXiv:2010.08120*.

[26] H. Araghi, M. Sabbaqi, and M. Babaie–Zadeh, "k-graphs: An algorithm for graph signal clustering and multiple graph learning," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1486–1490, Oct. 2019.

[27] H. Araghi, M. Babaie-Zadeh, and S. Achard, "Dynamic k-graphs: An algorithm for dynamic graph learning and temporal graph signal clustering," in *Proc. Eur. Signal Process. Conf.*, 2020, pp. 2195–2199.

[28] H. P. Maretic and P. Frossard, "Graph Laplacian mixture model," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 261–270, Mar. 2020.

[29] 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.

[30] A. Jung, "Learning the conditional independence structure of stationary time series: A multitask learning approach," *IEEE. Trans. Signal Process.*, vol. 63, no. 21, pp. 5677–5690, Nov. 2015.

[31] N. Tran, O. Abramenko, and A. Jung, "On the sample complexity of graphical model selection from non-stationary samples," *IEEE. Trans. Signal Process.*, vol. 68, pp. 17–32, Dec. 2019.

[32] A. Lotsi and E. Wit, "High dimensional sparse gaussian graphical mixture model," 2013, *arXiv:1308.3381*.

[33] C. Gao, Y. Zhu, X. Shen, and W. Pan, "Estimation of multiple networks in gaussian mixture models," *Electron. J. Stat.*, vol. 10, no. 1, pp. 1133–1154, May 2016.

[34] B. Hao, W. W. Sun, Y. Liu, and G. Cheng, "Simultaneous clustering and estimation of heterogeneous graphical models," *J. Mach. Learn. Res.*, vol. 18, no. 217, pp. 1–58, 2018.

[35] J. V. d. M. Cardoso and D. P. Palomar, "Learning undirected graphs in financial markets," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, 2020, pp. 741–745.

[36] R. Agrawal, U. Roy, and C. Uhler, "Covariance matrix estimation under total positivity for portfolio selection," *J. Financial Econometrics.*, vol. 20, pp. 367–389, Sep. 2020.

[37] S. Balakrishnan *et al.*, "Statistical guarantees for the EM algorithm: From population to sample-based analysis," *Ann. Stat.*, vol. 45, no. 1, pp. 77–120, Feb. 2017.

[38] Z. Wang, Q. Gu, Y. Ning, and H. Liu, "High dimensional expectation-maximization algorithm: Statistical optimization and asymptotic normality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2521–2529.

[39] S. Negahban, B. Yu, M. J. Wainwright, and P. K. Ravikumar, "A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1348–1356.

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

[41] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Artif. Intell. Stat.*, 2016, pp. 920–929.

[42] J. Ma and L. Xu, "Asymptotic convergence properties of the EM algorithm with respect to the overlap in the mixture," *Neurocomputing*, vol. 68, no. 10, pp. 105–129, Oct. 2005.

[43] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Prob.*, 1967, pp. 281–297.

[44] P. Erdös and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.

[45] M. D. Craven, D. Freitag, D. McCallum, A. Mitchell, T. Nigam, and K. Slattery, "Learning to extract symbolic knowledge from the world wide web," in *Proc. Nat. Conf. Artif. Intell.*, 1998, pp. 509–516.

[46] A. Cardoso-Cachopo, "Improving methods for single-label text categorization," Ph.D. dissertation, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, Lisbon, Portugal, 2007.

[47] S. T. Dumais, "Improving the retrieval of information from external sources," *Behav. Res. Methods Instrum. Comput.*, vol. 23, no. 2, pp. 229–236, Jun. 1991.

[48] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[49] S. Hassan-Moghaddam, N. K. Dhingra, and M. R. Jovanović, "Topology identification of undirected consensus networks via sparse inverse covariance estimation," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 4624–4629.

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

Yanli Yuan received the B.Sc. and M.Sc degrees in electrical and electronics engineering from the Beihang University of China, Beijing, China, in 2012 and 2015, respectively, and the Ph.D. degree in electronic engineering from the Singapore University of Technology and Design, Singapore, in 2021. She is currently a Postdoctoral Research Fellow with SUTD. Her research interests include network topology inference, graph learning, and statistical signal processing.

De Wen Soh received the B.S. degree in mathematics from Stanford University, Stanford, CA, USA, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, under the supervision of Sekhar Tatikonda, where he worked on high-dimensional graphical model learning. He is currently an Assistant Professor with the Singapore University of Technology and Design, Singapore. His research interests include graphical model estimation, graph signal processing, network analytics, transport modeling, high-dimensional statistical theory, and artificial intelligence.

Xiao Yang received the Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2021. Her research interests include wireless communications and networks, data science, network science, graph signal processing, and network topology inference.

Kun Guo (Member, IEEE) received the B.E. degree in telecommunications engineering and the Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2012 and 2019, respectively. From 2019 to 2021, she was a Postdoctoral Research Fellow with the Singapore University of Technology and Design, Singapore. She is currently a Zijiang Young Scholar with the School of Communications and Electronics Engineering, East China Normal University, Shanghai, China. Her research interests include edge computing, caching, and intelligence.

Tony Q.S. Quek (Fellow, IEEE) received the B.E. and M.E. degrees in electrical and electronics engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. He is currently the Cheng Tsang Man Chair Professor with the Singapore University of Technology and Design (SUTD), Singapore. He is also the Director of the Future Communications R&D Programme, Head of ISTD Pillar, and Deputy Director of the SUTD-ZJU IDEA. His research interests include wireless communications and networking, network intelligence, Internet of Things, URLLC, and 6G. Dr. Quek is actively involved in organizing and chairing sessions, and has was a member of the Technical Program Committee and symposium chairs in a number of international conferences. He is currently the Area Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and an elected member of the IEEE Signal Processing Society SPCOM Technical Committee. He was an Executive Editorial Committee Member of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE WIRELESS COMMUNICATIONS LETTERS. Dr. Quek was honored with the 2008 Philip Yeo Prize for Outstanding Achievement in Research, 2012 IEEE William R. Bennett Prize, 2015 SUTD Outstanding Education Awards – Excellence in Research, 2016 IEEE Signal Processing Society Young Author Best Paper Award, 2017 CTTC Early Achievement Award, 2017 IEEE ComSoc AP Outstanding Paper Award, 2020 IEEE Communications Society Young Author Best Paper Award, 2020 IEEE Stephen O. Rice Prize, 2020 Nokia Visiting Professor, and 2016–2020 Clarivate Analytics Highly Cited Researcher.