Online Inference for Mixture Model of Streaming Graph Signals With Sparse Excitation

Yiran He, Graduate Student Member, IEEE, and Hoi-To Wai, Member, IEEE

Abstract—This paper considers a joint multi-graph inference and clustering problem for simultaneous inference of node centrality and association of graph signals with their graphs. We study a mixture model of filtered low pass graph signals with excitation driven by a sparse matrix. While the mixture model is motivated from practical scenarios, it presents significant challenges to prior graph learning methods. As a remedy, we consider an inference problem focusing on the node centrality of graphs. We design an expectation-maximization (EM) algorithm with a unique low-rank plus sparse prior derived from low pass signal property. We propose a novel online EM algorithm for inference from streaming data. As an example, we extend the online algorithm to detect if the signals are generated from an abnormal graph. We show that the proposed algorithms converge to a stationary point of the maximum-a-posterior (MAP) problem. Numerical experiments support our analysis.

Index Terms—Blind centrality inference, clustering of graph signals, expectation maximization, online graph learning.

I. INTRODUCTION

The increasing demands for extracting information from complex systems have motivated the study of graphical models in many disciplines such as social science, biology, and data science. To analyze graph signals, i.e., observations made on the nodes, graph signal processing (GSP) [2], [3], [4], [5] has emerged as a natural framework for signal processing applications such as denoising [6], sampling [7], etc., also see the references in the above papers. Importantly, studies on graph topology learning using graph signal observations have been reported. Popular methods are proposed based on smoothness [8], [9], spectral template [10], topological constraints [11], causal modeling [12], nonlinear model [13], [14], partial observations [15]; see the overview papers [16], [17]. Moreover, a recent direction is to perform end-to-end learning for features of graph topology. The subjects of interest include centrality [18], [19], communities [20], [21], network processes [22], [23], etc. Compared to traditional graph learning, the latter approaches are usually more robust to challenging but realistic scenarios such as when the excitation is not white across nodes, or when the graph signals are not sufficiently smooth.

Many existing results on graph learning focus on a setting where the goal is to infer a single graph from data. In reality, the data can be more complex and is related to multiple graphs. For example, recent works [24], [25], [26], [27], [28], [29] studied the time varying graph learning problem when the topology changes slowly. Alternatively, one also considers the scenario where the graph topology differs significantly across samples. For example, a series of resting state brain networks have been identified from brain signals [30]; stock prices recorded at different states of the market may lead to different graph topologies [31]. This model is also relevant to the problem of detecting topology changes in graph signals [32], [33], [34], [35].

This paper treats a joint multi-graph centrality inference and clustering problem which simultaneously infers the node centrality of multiple graphs and clusters observed graph signals with respect to the graph that generates them. Our problem is motivated by applications involving multiple graphs with unknown associations between the graphs and observations. For example, when observing brain signals, we do not know which state the subject is in; for stock prices observations, the states of the market can be difficult to identify. While centrality inference can be performed by prior works [18], [19], the graph signal clustering problem is more challenging as classical algorithms such as spectral clustering [36], KNN [37] do not consider structures in the graph signal observations which is crucial to providing reliable estimates. Recent works have developed algorithms on simultaneous graph signal clustering and graph topology learning, e.g., graph Laplacian mixture model [38] and its regularized version [39], regularized spectral clustering [40], K-means based method [41]. Most of these works are developed from the Gaussian Markov random field model and entail stringent conditions such as requiring the observations to be generated from full-rank, white excitation across nodes. In comparison, our approach handles a relaxed mixture model of graph signal with possibly low-rank, non-white excitation. We remark that our objective of graph signal clustering differs from that of recent works on joint graph clustering and learning [42], [43], [44], [45]. These works assumed a single graph generative model where nodes in the unknown graph can be grouped into several densely connected clusters, while our model assumes a multi-graph generative model; see the illustration in Fig. 1.

The current paper also proposes an online algorithm for the joint inference problem with streaming data. Notice that many
existing graph learning algorithms require batch data. This is in contrast to the practical environment that involves streaming and even dynamical data collection. Furthermore, the online algorithm enjoys a low memory footprint and computation complexity by processing data on-the-fly. Several online algorithms on graph topology learning have been proposed, e.g., for time varying graph learning [46], [47], [48], for multi-graph topology learning but with pre-clustered data [49]. In contrast, our algorithm is the first to perform multi-graph inference and clustering simultaneously and in an online fashion. Our key contributions are:

- To study graph signal observations based on multiple graphs, we propose a mixture model of graph signals with general non-white excitation. Moreover, the model supports missing data and general observation model such as the logit model for inference from binary data.
- We formulate a joint inference and clustering problem via the MAP framework to infer node centrality and cluster graph signals according to the graphs. We design a batch EM algorithm under a unique low-rank plus sparse prior that adapts to the low pass signal property.
- We develop a novel online EM algorithm based on the stochastic approximation (SA) scheme for streaming data. The stochastic algorithm processes each of the incoming observation on-the-fly and features a low memory footprint while delivering similar performances as the batch EM. Our analysis shows that any fixed point of the algorithm is a stationary point of MAP. We also describe an application of the online algorithm to blind anomaly detection.
- We perform numerical experiments on synthetic and real data from brain and stock markets. The efficacy of the proposed algorithms supports our findings.

Compared to the conference version [1], this paper considers an extended signal model with missing data and logit observations. We also propose an online algorithm for streaming data and provide an extended set of experiments.

**Organization:** This paper is organized as follows. In Section II, we describe the mixture model of graph signals and then formally introduce the joint inference problem. Furthermore, we develop a maximum-a-posteriori formulation with low-rank plus sparse prior that adapts to the low pass signal property. In Section III, we propose a batch EM algorithm for Gaussian and logit observations. In Section IV, we introduce an online EM algorithm for streaming data and discuss its application to online anomaly detection. Finally, numerical experiments are presented to support our findings in Section V.

**Notations:** We use boldfaced character (resp. boldfaced capital letter) to denote vector (resp. matrix). For any vector $x \in \mathbb{R}^n$, $\|x\|_1$, $\|x\|_2$ denote the Euclidean, $\ell_1$ norm, respectively. For any matrix $X \in \mathbb{R}^{m \times n}$, we take $[X]_{i,j}$ to denote its $(i,j)$th entry. $\|X\|$, $\|X\|_1$, $\|X\|_*$ denote the spectral norm, $1$-norm, nuclear norm, respectively. We use $U(\cdot)$, $N(\cdot)$ to denote the uniform distribution, the normal distribution, respectively.

### II. Problem Statement

Consider $C$ undirected graphs $G^{(c)} = (\mathcal{V}, \mathcal{E}^{(c)}, A^{(c)})$, $c = 1, \ldots, C$. They share the same node set $\mathcal{V} := \{1, \ldots, n\}$ but with different edge sets $\{\mathcal{E}^{(c)}\}_{c=1}^C \subseteq \mathcal{V} \times \mathcal{V}$. Each graph $G^{(c)}$ is endowed with a symmetric weighted adjacency matrix $A^{(c)} \in \mathbb{R}^{n \times n}$ where $A^{(c)}_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}^{(c)}$; otherwise, $A^{(c)}_{ij} = 0$. Define the eigenvalue decomposition (EVD) $A^{(c)} = V^{(c)} \Lambda^{(c)} V^{(c)^\top}$. Each graph $G^{(c)}$’s centrality vector is given by the top eigenvector $v^{(c)}_1$. Node $i$ is said to be more central if the magnitude of its centrality value is greater.

We observe the graph signals on $\mathcal{V}$ generated from a process defined on one of the graphs. These graph processes are described via the linear graph filters [2]: for $c = 1, \ldots, C$,

$$
\mathcal{H}(A^{(c)}) = \sum_{r=0}^{P-1} h_r [A^{(c)}]^r \in \mathbb{R}^{n \times n},
$$

where $\{h_r\}_{r=0}^{P-1}$ are the filter coefficients, $P \in \mathbb{Z}_+ \cup \{\infty\}$ is the filter’s order. Each observation is indexed by $t \in \mathbb{Z}_+$ and is modeled as a graph signal matched with an identifier variable $w_t \in \{1, \ldots, C\}$. The latter indicates that the graph signal is generated from $G^{(w_t)}$. We describe the observation via a mixture model of graph signals with missing data:

$$
y_t = \Omega_t \odot \left\{ \sum_{c=1}^C 1(w_t = c) \mathcal{H}(A^{(c)}) x_t + e_t \right\},
$$

(2a)

$$
x_t = B z_t = \sum_{j=1}^r b_j [z_t]_j.
$$

(2b)

In (2a), $x_t \in \mathbb{R}^n$ is the excitation to the graph filter to be described later, $e_t \sim N(0, \sigma^2 I)$ is an independent and identically distributed (i.i.d.) Gaussian observation noise, $1(\cdot)$ is the $\{0,1\}$ indicator function, and $\odot$ is the element-wise product. The vector $\mathcal{H}(A^{(c)}) x_t$ is the output of the graph filter $\mathcal{H}(A^{(c)})$ with the excitation $x_t$. The graph identifier $w_t$ is an i.i.d. multinomial...
random variable (r.v.) with probability mass function $P(w_t = c) = P_c$. The binary vector $\Omega_t \in \{0,1\}^n$ models on which nodes the signal values are missing in the current sample. An extension to logit observations will be described in Section III-A.

In (2b), we further model that the excitation signal $x_t$ lies in a general $r$-dimensional subspace $\text{span}\{B\}$ with $B \in \mathbb{R}^{n \times r}$, $r \leq n$. The setting is in line with real world observations as data tends to be low-rank [50]. The vector $z_t \in \mathbb{R}^n$ is an excitation parameter whose element represents an observable source of stimuli on the graph process (can be deterministic or i.i.d. random). Each column $b_j$ is the influence profile from the $j$th source $z_{t,j}$ on the node set $V$. Specifically, we model $B$ as a sparse matrix which includes the special case of $B = I$. Fig. 2 summarizes the generation process\footnote{We remark that it is easy to extend (2) to the setting that every graph filter has different filter coefficients, every graph is associated with a different excitation subspace matrix $B^{(c)}$, etc.} of (2).

By default, we do not impose any assumption on the sequence $\{z_t\}_{t \geq 0}$. In the special case if the latter is i.i.d. random, then the observed sequence $\{y_t\}_{t \geq 0}$ would be i.i.d. However, even so, the fact that $B$ can be non-identity when $r < n$ has made it challenging, if not impossible, to perform inference on (2) such as reconstructing the graph topology $A^{(c)}$ from the filtered graph signals. Note that in the single graph setting, prior methods [8], [9], [10], [11] require the graph filter to be excited by a white noise, i.e., with $\mathbb{E}[x_t x'_t] = I$. As a remedy inspired by [18], [19], [20], [21], we aim to perform partial inference on (2) via the joint multi-graph centrality inference and graph signal clustering problem:

**Problem 1:** Given the data tuples $\{\text{DP}_t\}_t := \{y_t, z_t, \Omega_t\}_t$ from (2), estimate (A) eigen-centrality vector $v^{(c)}_{1\cdot}$ for each graph, and (B) identifier variable $w_t \in \{1, \ldots, C\}$ for each sample (subject to permutation ambiguity).

See Fig. 1. We consider two settings of data availability. In the first setting, the data tuples are available in a complete batch, i.e., one observes $\{\text{DP}_t\}_{t=1}^m$ where $m$ denotes the total number of samples. In the second setting, the data tuples are revealed in a streaming fashion. At time $t$, we only observe a sample $\text{DP}_t$ that is generated randomly according to (2). To avoid degeneracy, we assume that different graphs are equipped with different sets of central nodes such that $v^{(c)}_1 \neq v^{(c)}_1, c \neq c'$. We concentrate on graphs with small groups of central nodes of high-intra and low-inter connectivity. Such graphs typically admit a core–periphery structure which can be characterized by the eigengap condition $\lambda^{(c)}_1 \gg \lambda^{(c)}_2$ [51].

Tackling the joint inference problem is challenging due to the large number of unknowns in the model (2). For instance, even with $C = 1$, inferring the eigen-centrality vector $v^{(1)}_{1\cdot}$ from (2) is difficult since the graph filter $\mathcal{H}(A^{(1)})$, the excitation subspace $B$, etc., are unknown.

**Remark 1:** The requirement for excitation parameters $\{z_t\}_t$ to be known may appear restrictive at the first glance. However, we observe that in several applications where the role of excitation is clear from the context, estimates of these parameters can be obtained. For example,

- The parameters may be treated as the external sources that are directly observable as endogenous graph signals. In this case, $\{z_t\}_t$ can be obtained by observations on a subset of nodes whose signal values are extreme.
- The parameters may be estimated from side information known about the network. For instance, the excitation to stock networks can be modeled through the market’s interest level on various topics. A good surrogate for the latter can be given by the popularity of various keywords obtained from Google Trend.

See Section V-C for two example applications using real data.

### A. MAP Estimation With Reparametrization

This sub-section proposes a reparameterization technique to leverage the signal structure for finding a robust solution to Problem 1. We then formulate the maximum-a-priori (MAP) problem which will be the focus for the rest of this paper.

In the absence of knowledge on graph filters, Problem 1 will be ill-defined due to difficulty in extracting $v^{(c)}_{1\cdot}$. Taking inspirations from [18], [19], we consider a low pass assumption [2], [52] on the underlying graph filters:

**Assumption 1:** The graph filter $\mathcal{H}(A^{(c)})$, is 1-low pass with:

$$\eta^{(c)} := \max_{j=2, \ldots, n} |h(\lambda^{(c)}_1)/h(\lambda^{(c)}_2)| < 1, \quad (3)$$

for $c = 1, \ldots, C$, where the polynomial $h(\lambda) := \sum_{\tau=0}^{p-1} h_\tau \lambda^\tau$ is the frequency response of the graph filter $\mathcal{H}(\cdot)$.

The low pass ratio $\eta^{(c)}$ characterizes the strength of $\mathcal{H}(A^{(c)})$. With a smaller $\eta^{(c)}$, the filter $\mathcal{H}(A^{(c)})$ attenuates the signal components beyond the cutoff frequency $\lambda^{(c)}_2$ more. If $\eta^{(c)} \approx 1$, then $\mathcal{H}(A^{(c)})$ is considered as weak low pass; if $\eta^{(c)} < 1$, then $\mathcal{H}(A^{(c)})$ is considered as strong low pass. Assumption 1 is common in modeling network processes. Examples include, but are not limited to, opinion dynamics in social networks, stock dynamics, power systems, etc., see [52].

Assumption 1 implies that the top eigenvector of $\mathcal{H}(A^{(c)})$ corresponds to the centrality vector $v^{(c)}_{1\cdot}$. Together with the condition that $v^{(c)}_1 \neq v^{(c')}_1$, one may tackle Problem 1 through grouping the $m(\gg C)$ observations into $C$ clusters using naive spectral clustering. Particularly, assume that $\Omega_t = 1$, the $(t, t')$th element of the correlation matrix of observations is

$$\langle y_t | y_{t'} \rangle \approx \mathcal{H}(A^{(w_t)})B z_t | \mathcal{H}(A^{(w_{t'})})B z_{t'}$$  \quad (4)
Since the top eigenvectors of $H(A(w_{ij}))$, $H(A(w_{ij}'))$ differ only if $w_{ij} \neq w_{ij}'$, the $m \times m$ correlation matrix shall exhibit a block structure aligned with the graph identifiers $\{w_{ij}\}_{t=1}^n$.

To this end, an intuitive idea is to apply spectral clustering (SC) on the correlation matrix $\hat{C}_Y = [y_1, \ldots, y_m]^\top[y_1, \ldots, y_m]$ to cluster the graph signals. However, as demonstrated below, the result is sensitive to the low pass filter modeling the graph process:

Example 1: We generate $C = 2$ core-periphery graphs with $n = 100$ nodes, each with $10$ distinct central nodes that are fully connected, and $m = 400C$ graph signals are generated according to (2). Fig. 3 shows the scatter plot whose coordinates of the data from two core-periphery graphs. (Left) Strong low pass filter $H_w(A) = (I - \frac{1}{m} A)^{-1}$; (Right) Weak low pass filter $H_w(A) = (I - \frac{1}{m} A)^{-1}$.

![Fig. 3. Toy Example illustrating the data from two core periphery graphs. (Left) Strong low pass filter $H_w(A) = (I - \frac{1}{m} A)^{-1}$; (Right) Weak low pass filter $H_w(A) = (I - \frac{1}{m} A)^{-1}$.](image)

The naive SC essentially utilizes difference in the subspaces $\text{span}(H(A(c))B)$, $c = 1, \ldots, C$ to discern samples from different graphs. While such strategy is successful when $H(A(c))$ is strong low pass, it may not work when the filter is weak low pass; see Remark 2 for further justifications. Nevertheless, the above example shows that inferring the natural parameters $\{H(A(c))B\}_{c=1}^C$ can be insufficient for a robust solution to Problem 1.

Particularly, the above example demonstrates that it is necessary to jointly consider the signal structure while clustering the graph signals. Our idea is to model and extract the hidden component(s) in $H(A(c))B$ that are indicative of the eigen-centrality vector, which thus provides the graph identifiers necessary for clustering. Observe the decomposition:

$$H(A(c))B = (H(A(c)) - \rho I)B + \rho B = L_c + B_p, \quad (5)$$

for any $\rho \geq 0$. The component $L_c = (H(A(c)) - \rho I)B$ depends on the shifted graph filter $H(A(c)) - \rho I$. It is shown [20, Observation 1] that there exists $\rho > 0$ where the shifted graph filter enjoys a strictly lower low pass ratio, denoted as $\hat{\eta}(c)$, than the original ratio $\eta(c)$. For example, with $H(A(c)) = (I - \alpha A(c))^{-1}$, it can be shown that the shifted graph filter with $\rho = 1$ has the low pass ratio of $\hat{\eta}(c) \leq \frac{\lambda_2}{\lambda_1} \eta(c) \ll \eta(c)$, provided that $\lambda_1 \gg \lambda_2$ which can be satisfied for graphs with core-periphery structure [51]. Consequently, the matrix $L_c$ will be approximately rank-one.

Below, we show that the low-rank components $L_c$ are distinct for the different graphs that they are associated with. Assume without loss of generality (w.l.o.g.) that $\hat{\eta}(c) \geq 0$, the following lemma is adapted from [19, Corollary 1]:

**Lemma 1:** For each $c = 1, \ldots, C$, if $(v_1^{(c)})^\top B q_1^{(c)} \neq 0$, then

$$\|v_1^{(c)} - \hat{v}_1^{(c)}\| \leq \sqrt{2\hat{\eta}(c)} \frac{\|V_n^{(c)}\|}{\|v_1^{(c)}\|}, \quad (6)$$

where $V_n^{(c)}$ is the last $n - 1$ eigenvectors of $A(c)$, and $\hat{v}_1^{(c)}, q_1^{(c)}$ are the top left, right singular vector of $L_c$.

The right hand side of (6) is bounded by $O(\hat{\eta}(c))$ with $\hat{\eta}(c) \ll 1$. Together with the observation that $L_c$ is approximately rank one, we obtain $L_c \propto v_1^{(c)} \circ (q_1^{(c)})^\top$. With the condition $v_1^{(c)} \neq \hat{v}_1^{(c)}$, we observe that $L_c$ provides an effective indicator to distinguish the samples with different graph identifiers.

Establishing that $L_c$ is low rank may be insufficient for its recovery in (5), where extra structure has to be leveraged for the residual term $\rho B$ as inspired by [53]. Fortunately, since $B$ models the influences from external sources on the graph(s), we note from the applications described in [19], [20] that $B$ admits certain low-dimensional structure. For example, $B$ can be sparse, the number of non-zero row/column vectors of $B$ can be small, etc. Herein, we recall the model that $B$ is a sparse matrix. We observe that the matrix product $H(A(c))B$ admits a ‘low-rank plus sparse’ structure under the said premises.

**MAP Estimation:** The above motivates us to explicitly account for the implicit components $L_c, B_p$ during the inference process through a careful re-parameterization. Denote $\Theta := \{L_c\}_{c=1}^C, B_p, \{P_c\}_{c=1}^C$ as the set of parameters. We yield the following structured MAP estimation problem:

$$\max_{\Theta \in \mathcal{Z}} L(\Theta) := \log p(Y|\Theta, Z, \Omega) + \log p(\Theta), \quad (7)$$

where $\mathcal{Z} = \{\Theta : P_c \geq 0, \sum_{c=1}^C P_c = 1\}$ and $p(\Theta)$ models the prior on $\Theta$ with the ‘low-rank plus sparse’ structure of $L_c, B_p$.

A natural choice for the prior distribution is

$$p(\Theta) \propto \exp \left(-\lambda_S \|B_p\|_1 - \lambda_L \sum_{c=1}^C \|L_c\|_*, \right), \quad (8)$$

where $\lambda_S, \lambda_L \geq 0$ are regularization parameters. Furthermore, the expectation $\mathbb{E}[\cdot]$ is defined w.r.t. the observation law for $Y, Z, \Omega$ and the log-likelihood function is given by:

$$\log p(y|\Theta, z, \Omega) = \log \left(\sum_{c=1}^C \sum_{e \in E_c} e^{-\frac{1}{\eta(c)} f_e(|y - \Omega_{\Theta}(L_c + B_p)z|)}\right) + \text{constant}, \quad (9)$$

which is non-concave due to the nonlinear coupling between $P_c, L_c, B_p$, making direct optimization of (7) intractable.
We notice that EM algorithms for classical models such as Gaussian Mixture Model (GMM) cannot be directly applied to (7). The reason is that (7) entails regularization terms for the ‘low-rank plus sparse’ structure. Additionally, the missing data and excitation parameter $z_t$ have to be incorporated into the inference process. In the next section, we concentrate on developing effective algorithms for (7) via the EM paradigm.

Remark 2: The poor performance of naïve SC with weak low pass filter can be explained by [19, Lemma 2]. Under mild conditions, the latter lemma shows:

$$\|L_c\|/\|B_\rho\| \lesssim (1 - \eta(c))/(1 + \eta(c)),$$

(10)

where $L_c, B_\rho$ are defined in (5). Now, if $\eta(c) \approx 1$, we have $\mathcal{H}(A^{(c)}) B \approx B_\rho$ for any $c = 1, \ldots, C$. Consequently, the correlation matrix $\tilde{C}_Y \approx [z_1, \ldots, z_m]^T B_\rho^T B_\rho [z_1, \ldots, z_m]$ does not have the anticipated block structure that is necessary for successful clustering.

### III. BATCH EM ALGORITHM

This section develops a customized EM algorithm that is efficient to implement and enjoys desirable theoretical properties. Particularly, we focus on batch data where $m$ samples $\{DP_t\}_{t=1}^m$ are available all at once.

To begin, let us fix $\tilde{\Theta} = \{\{\tilde{L}_c\}_{c=1}^C, \tilde{\bar{B}}_\rho, \{\tilde{P}_c\}_{c=1}^C\}$ and denote the conditional probability mass function for the latent r.v. $w_t$ as $q(\cdot|\tilde{\Theta}, DP_t)$. The Jensen’s inequality implies the following lower bound on the log-likelihood term in (7):

$$\log p(y_t|\tilde{\Theta}, z_t, \Omega_t) = \log \left( \sum_{c=1}^C p(y_t, w_t = c|\tilde{\Theta}, z_t, \Omega_t) \frac{q(w_t = c|\tilde{\Theta}, DP_t)}{q(w_t = c|\tilde{\Theta}, DP_t)} \right) \geq \mathbb{E}_{w_t \sim q(\cdot|\tilde{\Theta}, DP_t)}[\log p(y_t, w_t|\tilde{\Theta}, z_t, \Omega_t)] + \zeta(\tilde{\Theta}),$$

(11)

where $\zeta(\tilde{\Theta})$ is a function that only depends on the fixed $\tilde{\Theta}$. Taking the batch data setting into consideration and the assumption that $w_t, e_t$ are i.i.d. over $t$, the above led us to the batch surrogate optimization problem:

$$\max_{\tilde{\Theta}} \frac{1}{m} \sum_{t=1}^m \mathbb{E}_{w_t \sim q(\cdot|\tilde{\Theta}, DP_t)}[\log p(y_t, w_t|\tilde{\Theta}, z_t, \Omega_t)] - \lambda_S \|B_\rho\|_1 - \lambda_L \sum_{c=1}^C \|L_c\|_*$

s.t. $\sum_{c=1}^C P_c = 1, P_c \geq 0, c = 1, \ldots, C.$

(12)

Let us take a closer look at the first term in the objective function of (12). For $t = 1, \ldots, m$, the Bayes’ rule implies

$$\mathbb{E}_{w_t \sim q(\cdot|\tilde{\Theta}, DP_t)}[\log p(y_t, w_t|\tilde{\Theta}, z_t, \Omega_t)] = \mathbb{E}_{w_t|\{w_t = c\}}[\log \mathbb{P}(w_t = c)p(y_t|\tilde{\Theta}, z_t, \Omega_t, w_t = c)] = \mathbb{E}_{w_t|\{w_t = c\}}[\{\log(P_c) + \log p(y_t|B_\rho, L_c, z_t, \Omega_t)\}],$$

where we used $\mathbb{P}(w_t = c) = P_c$ and the dependence on $q(\cdot|\tilde{\Theta}, DP_t)$ were omitted for brevity. Moreover,

$$\log p(y_t|B_\rho, L_c, z_t, \Omega_t) = \frac{\langle \Omega_t \cdot (L_c + B_\rho)z_t|y_t \rangle}{\sigma^2} - \frac{\|\Omega_t \cdot (L_c + B_\rho)z_t\|^2}{2\sigma^2} + \tilde{\zeta}(\tilde{\Theta}),$$

(13)

where $\tilde{\zeta}(\tilde{\Theta}) := \frac{\zeta(\tilde{\Theta})}{\sigma^2} \|y_t\|^2$. The above expressions can be simplified as

$$\langle \Omega_t \cdot (L_c + B_\rho)z_t|y_t \rangle = \text{Tr}((L_c + B_\rho)^\top \Omega_t \cdot y_tz_t^\top),$$

$$\|\Omega_t \cdot (L_c + B_\rho)z_t\|^2 = \sum_{i=1}^n \text{Tr}((L_c + B_\rho)^\top e_ie_i^\top (L_c + B_\rho) \Omega_t \cdot z_tz_t^\top).$$

Define the conditional probability for the event that the $t$th data tuple is associated to the $c$th graph:

$$p_c(\tilde{\Theta}, DP_t) := \mathbb{E}_{w_t \sim q(\cdot|\tilde{\Theta}, DP_t)}[1(w_t = c)] = \frac{P_c \exp\left(\frac{-1}{\sigma^2} \|y_t - \tilde{\Omega}_t \cdot (\tilde{B}_\rho + \tilde{L}_c)z_t\|^2\right)}{\sum_{c'=1}^C P_{c'} \exp\left(\frac{-1}{\sigma^2} \|y_t - \tilde{\Omega}_t \cdot (\tilde{B}_\rho + \tilde{L}_{c'}z_t)\|^2\right)},$$

(14)

and the sufficient statistics:

$$\mathbb{Y}Z_c = \frac{1}{m} \sum_{t=1}^m p_c(\tilde{\Theta}, DP_t) \Omega_t \cdot y_tz_t^\top,$$

$$\mathbb{Z}Z_{c,i} = \frac{1}{m} \sum_{t=1}^m p_c(\tilde{\Theta}, DP_t) \Omega_t \cdot z_tz_t^\top.$$

(15)

For any $\tilde{\Theta} \in \mathcal{T}$, the lower bound surrogate objective function of (12) can be written as

$$\tilde{\mathcal{L}}(\Theta|\tilde{\Theta}) = \tilde{\zeta}(\tilde{\Theta}) - \lambda_S \|B_\rho\|_1 - \sum_{c=1}^C \lambda_L \|L_c\|_* + \sum_{c=1}^C \left\{ P_c \log(P_c) + \frac{\lambda_L}{\sigma^2} \text{Tr}((L_c + B_\rho)^\top \mathbb{Y}Z_c^2) \right\} - \frac{1}{2\sigma^2} \sum_{i=1}^n \text{Tr}((L_c + B_\rho)^\top e_ie_i^\top (L_c + B_\rho) \mathbb{Z}Z_{c,i}^2),$$

(16)

which is a concave function in $\Theta$ and we have defined $\tilde{\zeta}(\tilde{\Theta}) = (1/m) \sum_{t=1}^m \zeta(\tilde{\Theta})$. For any $\Theta, \tilde{\Theta} \in \mathcal{T}$, notice that it holds

$$\mathcal{L}(\Theta) \geq \tilde{\mathcal{L}}(\Theta|\tilde{\Theta}), \quad \mathcal{L}(\tilde{\Theta}) = \tilde{\mathcal{L}}(\Theta|\tilde{\Theta}).$$

(17)

The above derivations led us to a regularized (batch) EM algorithm. In particular, we initialize by fixing $\tilde{\Theta}^0$ and evaluate the sufficient statistics using (15). Then, we alternate between the M-step and the E-step — in the M-step, we optimize w.r.t. $\tilde{\Theta}$ for the surrogate problem (12); in the E-step, we update the sufficient statistics using (15) through the new $\tilde{\Theta}$. The overall algorithm is summarized in Algorithm 1.
Algorithm 1: Batch EM for Partial Inference on (2).

1: **Input:** graph signals $Y$, excitation parameters $Z$, missing-information vectors $\{\Omega_t\}_{t=1}^m$, no. of graphs $C$.
2: if $\Theta^0 := \{\{L_c\}_c, B^0_c, \{P^0\}_c\}$ is available then
3: Evaluate (14), (15).
4: else if $p_c(\Theta^0, D_P), t = 1, \ldots, m$ are available then
5: Evaluate (15).
6: end if
7: for $k = 1, 2, \ldots, K_{\text{max}}$ do
8: **M-step:** solve the concave maximization
   \[
   \Theta^k \in \arg \max_{\Theta} \mathcal{L}(\Theta | \Theta^{k-1})
   \]
   s.t. \[
   \sum_{c=1}^C P_c = 1, P_c \geq 0, \forall c,
   \]
   see (24) for efficient implementation in Gaussian case.
9: **E-step:** evaluate $p_c(\Theta^k, D_P)$ for all $t$ using (14), and the sufficient statistics using (15).
10: end for
11: **Output:** converged parameters $\Theta^{K_{\text{max}}}$ and conditional probabilities $p_c(\Theta^{K_{\text{max}}}, D_P)$ for all $t$.

Denote $\mathcal{D}(\Theta | \Theta) := \mathcal{L}(\Theta) - \mathcal{L}(\Theta | \tilde{\Theta}) \geq 0$ as the difference function between $\mathcal{L}(\Theta)$ and the surrogate. We have:

**Proposition 1:** Consider the sequence $\{\Theta^k\}_{k \geq 0}$ generated by Algorithm 1. The following holds:
1) The regularized log-likelihood value is non-decreasing:
   \[
   \mathcal{L}(\Theta^{k+1}) \geq \mathcal{L}(\Theta^k), \forall k \geq 0.
   \]
2) If the gradient w.r.t. $\Theta$ for the difference function $\mathcal{D}(\Theta | \Theta)$ is $L$-Lipschitz continuous, then for any $K_{\text{max}} \geq 1$,
   \[
   \min_{k=1, \ldots, K_{\text{max}}} \| \nabla \mathcal{D}(\Theta^k | \Theta^{k-1}) \|^2 = O(L / K_{\text{max}}).
   \]
   In addition, the directional derivative:
   \[
   \mathcal{L}'(\tilde{\Theta}; \Theta - \tilde{\Theta}) := \lim_{\delta \to 0} \frac{\mathcal{L}(\tilde{\Theta} + \delta(\Theta - \tilde{\Theta})) - \mathcal{L}(\tilde{\Theta})}{\delta}
   \]
   exists for any $\Theta, \tilde{\Theta} \in \mathcal{I}$. Thus,
   \[
   \min_{k=1, \ldots, K_{\text{max}}} \sup_{\Theta \in \mathcal{I}} \mathcal{L}'(\Theta^k; \Theta - \Theta) / \|\Theta^k - \Theta\| = O\left(\sqrt{\frac{L}{K_{\text{max}}}}\right).
   \]
   Note that if $\mathcal{L}'(\tilde{\Theta}; \Theta - \tilde{\Theta}) \leq 0$ for all $\Theta \in \mathcal{I}$, then $\Theta$ is a stationary point to the MAP problem (7). As such, Algorithm 1 finds a stationary point to (7) at a sublinear rate. A key challenge in our analysis is that the MAP problem (7) is non-smooth due to the sparse/low-rank priors in (8). We achieve the proof through extending [54], [55], see Appendix A.

**Implementation Details:** We comment on the $M$-step. First, the maximizer for $\{P_c\}_c$ is given by:
\[
P_c = P_c^\Phi \left( \sum_{c=1}^C P_c^\Phi \right)^{-1}, \quad c = 1, \ldots, C.
\]
Second, the parameters $\{B_c, \{L_c\}_c\}$ can be obtained through solving the regularized least square problem:
\[
\min_{\{L_c\}_c, B_c} \lambda \sum_{c=1}^C \|L_c\|_1 + \gamma \|B_c\|_1 + \frac{1}{2\sigma^2} \left( \sum_{c=1}^C \sum_{i=1}^n \|e_i^T(L_c + B_c)Z_c^\Phi - e_i^T YZ_c^\Phi \|_2^2 \right)
\]
where $Z_c^\Phi$ is the matrix square root of $Z_cZ_c^\Phi$. Note (24) is a convex robust PCA problem [53] which can be efficiently solved by available software such as [53], [56]. At each iteration of Algorithm 1, the $E$-step (15) takes the computation complexity of $O(mnr)$. For the $M$-step, we update the common variable $B_c$ into $B_c^\Phi$, $c = 1, \ldots, C$ and (24) can be separated into $C$ decoupled regularized least square problems. Now using [56], the per-inner iteration computation complexity for solving these $C$ problems is $O(n^2r)$. Besides, Algorithm 1 also supports initialization in the absence of $\Theta^0$. In fact, it suffices to initialize the algorithm through evaluating the sufficient statistics in the $E$-step. For the latter, we estimate the conditional probability that the $t$th data tuple is associated to the $c$th graph, e.g. by applying the naive SC. The sufficient statistics can then be found using (15).

Finally, we demonstrate how to tackle Problem 1 using the outputs from Algorithm 1. The operations are straightforward:
(i) the eigen-centrality can be estimated by applying SVD on the matrices $\{L_c^{K_{\text{max}}}\}_c$ and extract the top left singular vectors; (ii) the graph identifiers are estimated by taking
\[
\hat{w}_t = \arg \max_{c=1, \ldots, C} p_c(\Theta^{K_{\text{max}}}, D_P),
\]
for all $t = 1, \ldots, m$.

A. Extension to Logit Model

We conclude this section by extending Algorithm 1 to tackling Problem 1 with binary graph signal observations. For example, this applies if the latter consists of vote data. Consider the case without missing data, i.e., $\Omega_t = 1$ and focus on a logit observation model. The observed data $\{D_P\}_t = \{y_t, z_t\}_t$ satisfy:
\[
P(y_{t,j} = Y) = \frac{\exp((\tilde{y}_{t,j} + b)Y)}{1 + \exp((\tilde{y}_{t,j} + b)Y)}, \quad Y \in \{0, 1\},
\]
for $j = 1, \ldots, n$, where $b < 0$ is the bias parameter of the logit model and $\tilde{y}_{t,j}$ is the $j$th element of the vector:
\[
\tilde{y}_t = \sum_{c=1}^C \mathbb{1}(w_t = c)H(A(c))Bz_t.
\]
Similar to Section II-A, we further adopt the parameterization with $H(A(c))B = L_c + B_c$. The EM algorithm on the above model can be developed similarly as Algorithm 1. In particular, the derivations up to (11) remain valid. Now, denote the conditional probability of
the graph identifier \( u_t \) [cf. (14)] for the \( t \)th data tuple as:

\[
p_{\hat{Y}}(\Theta, \text{DP}_t) = \frac{\bar{P}_{t} \prod_{j=1}^{n} \exp\left(y_{j}, \bar{\nu}_{t,j,c}\right)}{\sum_{c'=1}^{C} P_{c'} \prod_{j=1}^{n} \exp\left(y_{j}, \bar{\nu}_{t,j,c}'\right)},
\]

where \( \bar{\nu}_{t,j,c} := b + e_{j}^\top (L_{c} + B_{p}) z_{t} \). We observe that the following surrogate objective function lower bounds the MAP objective function with the logit model (26):

\[
\tilde{\mathcal{L}}_{\text{logit}}(\Theta \mid \Theta) = \text{constant} - \lambda_{S} ||B_{p}||_{1} - \sum_{c=1}^{C} \lambda_{L} ||L_{c}||_{*} + \frac{1}{m} \sum_{c=1}^{C} \sum_{t=1}^{m} p_{\hat{Y}}(\Theta, \text{DP}_t) \left\{ \log(P_{c}) + \sum_{j=1}^{n} y_{j} \nu_{t,j,c} \right\} - \frac{1}{m} \sum_{c=1}^{C} \sum_{t=1}^{m} p_{\hat{Y}}(\Theta, \text{DP}_t) \sum_{j=1}^{n} \log(1 + \exp(\nu_{t,j,c})),
\]

where \( \nu_{t,j,c} := b + e_{j}^\top (L_{c} + B_{p}) z_{t} \) is a linear function of the decision variables \( L_{c}, B_{p} \).

We observe that (29) is a concave function in \( \Theta \). To develop the EM algorithm, the M-step in Algorithm 1 can now be replaced by maximizing (29) w.r.t. \( \Theta \) when \( \Theta^{-1} \) is given. On the other hand, E-step only involves evaluating \( p_{\hat{Y}}(\Theta, \text{DP}_t) \) according to (28). Compared to the case with Gaussian observation, the M-step involves (29) which is a finite-sum problem that can be difficult to optimize when \( m \gg 1 \). This is caused by the nonlinear log-likelihood function associated with the logit model (26).

IV. ONLINE EM ALGORITHM

This section considers tackling Problem 1 under streaming data. We focus on an online learning process where the data tuple is revealed sequentially. Particularly, at time \( t \), we only observe the data tuple \( \text{DP}_{t} = \{y_{t}, z_{t}, \Omega_{t}\} \) that is generated from the model (2) in an i.i.d. fashion.

We aim to design an online algorithm for the MAP problem (7) with stochastic log-likelihood objective. Similar to (12), consider the following surrogate problem at the \( t \)th iteration:

\[
\max_{\Theta} \mathbb{E}_{\text{DP}, W \sim q(\Theta^{-1}, \text{DP})} \left[ \log p(Y, W \mid \Theta, Z, \Omega) \right] - \Psi(\Theta)
\]

with \( \Psi(\Theta) := \lambda_{S} ||B_{p}||_{1} + \lambda_{L,0} \sum_{c=1}^{C} ||L_{c}||_{*} + \epsilon \sum_{c=1}^{C-1} \log(P_{c}) + \epsilon \log\left(1 - \sum_{c=1}^{C-1} P_{c}\right),
\]

where the expectation \( \mathbb{E}_{\text{DP}} \) is taken w.r.t. the random generative model for the data tuple \( \text{DP} = \{Y, Z, \Omega\} \), and \( \lambda_{S}, \lambda_{L,0}, \epsilon > 0 \) are regularization parameters. Compared to (12), the additional regularizer on \( \{P_{c}\}_{c=1}^{C} \) enforces the latter to be in the interior of the simplex set.

The surrogate objective function of (30) admits a similar form as (16), which is derived as (constants are omitted)

\[
\tilde{\mathcal{L}}_{\text{al}}(\Theta; \tilde{P}_{c}, Y Z_{c}, \tilde{Z}_{c,i}, \forall c, i) := -\Psi(\Theta) + \sum_{c=1}^{C} \left\{ \tilde{P}_{c} \log(P_{c}) + \frac{1}{\sigma} \text{Tr} \left( (L_{c} + B_{p})^\top Y Z_{c}^{\Theta^{-1}} \right) \right\} - \frac{1}{2 \sigma^{2}} \sum_{i=1}^{n} \text{Tr} \left( (L_{c} + B_{p})^\top e_{i} e_{i}^\top (L_{c} + B_{p}) Z_{c,i}^{\Theta^{-1}} \right),
\]

where we have defined the population sufficient statistics as:

\[
\tilde{T}_{c}^{\Theta^{-1}} = \mathbb{E}_{\text{DP}}[p_{c}(\Theta^{-1}, \text{DP})],
\]

\[
\tilde{Z}_{c,i}^{\Theta^{-1}} = \mathbb{E}_{\text{DP}}[p_{c}(\Theta^{-1}, \text{DP}) \Omega_{i} Z_{c,i}^\top],
\]

\[
\tilde{Y} Z_{c}^{\Theta^{-1}} = \mathbb{E}_{\text{DP}}[p_{c}(\Theta^{-1}, \text{DP}) \Omega \odot Y Z_{c}^\top],
\]

such that \( p_{c}(\Theta^{-1}, \text{DP}) \) was defined in (14). Note that (31), (32) generalize (15), (16) to observations drawn from any distribution. To see this, (15) can be recovered from (32) as the special case with empirical distribution.

Following the development of the batch EM algorithm, we wish to maximize (31) w.r.t. \( \Theta \) at the M-step. However, unlike (15), computing (32) is challenging as we are observing the data tuple on-the-fly. To this end, we adopt the stochastic approximation (SA) scheme [57] from [58], [59] on the space of sufficient statistics to dynamically track (32).

SA Scheme for (32): Let \( \tilde{T}_{c}^{\tilde{T}-1}, \tilde{Y} Z_{c}^{\tilde{T}-1}, \tilde{Z}_{c,i}^{\tilde{T}-1} \) be the estimate for the sufficient statistics at iteration \( t - 1 \), we consider the following SA scheme to estimate (32):

\[
\tilde{T}_{c}^{t} = \tilde{T}_{c}^{\tilde{T}-1} + \beta_{t} \left( p_{c}(\Theta^{-1}, \text{DP}) - \tilde{T}_{c}^{\tilde{T}-1}\right),
\]

\[
\tilde{Y} Z_{c}^{t} = \tilde{Y} Z_{c}^{\tilde{T}-1} + \beta_{t} \left( p_{c}(\Theta^{-1}, \text{DP}) \Omega_{i} y_{i} z_{i}^\top - \tilde{Y} Z_{c}^{\tilde{T}-1}\right),
\]

\[
\tilde{Z}_{c,i}^{t} = \tilde{Z}_{c,i}^{\tilde{T}-1} + \beta_{t} \left( p_{c}(\Theta^{-1}, \text{DP}) \Omega_{i} z_{i} z_{i}^\top - \tilde{Z}_{c,i}^{\tilde{T}-1}\right),
\]

for \( i = 1, \ldots, n, c = 1, \ldots, C \), where \( \beta_{t} \in (0, 1] \) is the step size. Notice that the SA scheme only uses the current data \( \text{DP}_{t} \) available in the streaming data setting, where it replaces the E-step in the EM algorithm. The above estimates are then used in (31) to construct the surrogate \( \tilde{\mathcal{L}}_{\text{al}}(\Theta; \tilde{P}_{c}, Y Z_{c}, \tilde{Z}_{c,i}, \forall c, i) \), whose maximization leads to the M-step.

To understand (33), let us focus on \( \tilde{T}_{c}^{t} \) for the illustration purpose. Herein, the mean field of SA update is given by the expected value of the drift term conditioned on iterates up to the \( t - 1 \)th iteration. The latter is

\[
\mathbb{E}_{t-1} \left[ p_{c}(\Theta^{-1}, \text{DP}) - \tilde{T}_{c}^{t-1}\right] = \mathbb{E}_{\text{DP}} \left[ p_{c}(\Theta^{-1}, \text{DP}) \right] - \tilde{T}_{c}^{t-1}.
\]

Substituting into (32) shows that in expectation, \( \tilde{T}_{c}^{t} \) is a convex combination of \( \tilde{T}_{c}^{t-1} \) and \( \tilde{T}_{c}^{\tilde{T}-1} \). In other words, the recursion (33) drives the sufficient statistics estimates towards (32).
Algorithm 2 Online EM for Partial Inference on (2)

1: Input: no. of graphs $C$, initial parameters $\Theta^0$ and sufficient statistics $\{\overline{P}_c^{(i)}, \overline{Y}Z_c^{(i)}, \{\overline{Z}Z_c^{(i)}\}_i\}=1\to C$.
2: for $t = 1, 2, \ldots$ do
3: Sample: $DP_t = \{y_t, z_t, \Omega_t\}$ according to (2).
4: // Optional: Anomaly Detection
5: $\{\text{go to line 5, if (38) outputs } \mathcal{H}_0, \text{ go to line 3, if } \text{otherwise}\}$
6: E-step: update the sufficient statistics via (33).
7: M-step: maximize the surrogate function through solving

$$
\Theta^t \in \arg \max_{\Theta} \bar{L}_\epsilon(\Theta; \overline{P}_c^{(t)}, \overline{Y}Z_c^{(t)}, \overline{Z}Z_c^{(t)}, \forall c, i).
$$

8: end for

Equipped with the above derivations, we summarize the online EM algorithm in Algorithm 2. Note that the algorithm is fully online as it does not store the history of $\{DP_t\}_{t>0}$. Instead, information from the latter is absorbed by the sufficient statistics estimates in each iteration. Lastly, though Algorithm 2 bears similarities to [58], [59], our algorithm incorporates a set of non-smooth regularizers, i.e., $\|B_0\|_1, \|L_c\|_1$, that are motivated by the graph signal model. Note that Algorithm 2 has similar computation complexity for M-step as Algorithm 1, whereas the computational complexity of its E-step is $O(\text{mnr})$ which is reduced from $O(\text{mnwr})$ of Algorithm 1. The per-iteration complexity is independent of the dataset size.

Lastly, let us comment on the fixed point for the recursions (32). Note that a fixed point of $\{\overline{P}_c^{(t)}, \overline{Y}Z_c^{(t)}, \overline{Z}Z_c^{(t)}, \forall c, i\}$ for the recursion satisfies for any $c = 1, \ldots, C$ that

$$
\begin{align*}
E_{DP_c}[p_c(\overline{\Theta}, DP_c)] - \overline{P}_c &= 0, \\
E_{DP_c}[p_c(\overline{\Theta}, DP_c)\Omega_c \circ YZ_c] - \overline{Y}Z_c &= 0, \\
E_{DP_c}[p_c(\overline{\Theta}, DP_c)\Omega_i \circ ZZ_c] - \overline{Z}Z_{c,i} &= 0, \forall i,
\end{align*}
$$

(34)

where $\overline{\Theta} \in \arg \max_{\Theta} \bar{L}_\epsilon(\Theta; \overline{P}_c \circ \overline{Y}Z_c, \overline{Z}Z_{c,i}, \forall c, i)$. Observe:

**Proposition 2:** Let $\Gamma$ be the set of stationary solutions of the MAP problem with modified regularizer [cf. (7), (30)]:

$$
\max_{\Theta} E_{DP_c}[\log p(Y|\Theta, Z, \Omega)] - \Psi(\Theta),
$$

(35)

If the sufficient statistics $(\overline{P}_c, \overline{Y}Z_c, \overline{Z}Z_{c,i}, \forall c, i)$ satisfies (34), then $\overline{\Theta} \in \Gamma$. Conversely, assume in addition, the maximizer of $\bar{L}_\epsilon(\Theta; \overline{P}_c, \overline{Y}Z_c, \overline{Z}Z_{c,i}, \forall c, i)$ is unique for any sufficient statistics [cf. line 6 of Algorithm 2]. Then if $\overline{\Theta} \in \Gamma$, the tuple $(\overline{P}_c, \overline{Y}Z_c, \overline{Z}Z_{c,i}, \forall c, i)$ satisfies (34).

The proof, which extends [58], [60] to the regularized MAP setting in (35), is relegated to Appendix B.

The above proposition shows that if the SA recursion (33) converges to a fixed point, then such fixed point must lead to the parameter $\overline{\Theta}$ stationary to the MAP problem (35). The convergence of (33) to a fixed point typically requires

$$
\sum_{t=1}^{\infty} \beta_t = \infty, \sum_{t=1}^{\infty} \beta_t^2 < \infty,
$$

(36)

and additional conditions such as Lipschitz continuity of the population sufficient statistics map (32). In the interest of space, the readers are referred to [59], [61] for details. We remark that the stochastic gradient EM algorithm in [62] is an alternative to Algorithm 2. However, [62] applies stochastic gradient in the parameter ($\Theta$) space, which can be less computationally efficient.

A. Online Joint Inference & Anomaly Detection

We conclude by discussing an application of Algorithm 2 to online joint inference and anomaly detection of graph signals that are not generated from one of the candidate graphs, $G^{(c)}, c = 1, \ldots, C$, in (2). Detecting if graph signals are originated from an ‘abnormal’ graph is an important task for, e.g., power systems, pathological signal detection, see [32], [35]. While prior works require the normal graph topology to be known a-priori, our goal is to simultaneously perform graph inference through estimating central nodes and detect these abnormal graph signals. In this setting, an online algorithm is preferred as we aim to detect anomalies as soon as possible.

At time $t \geq 1$, we consider the graph signal $y_t$ (and the latent variable $z_t$) satisfying $y_t = \Omega_t \circ \mathcal{H}(A_t)B_z + e_t$ akin to (2). Herein, $A_t$ denotes the adjacency matrix of a graph $G_t$ that $y_t$ is originated from. Accordingly, $y_t$ is said to be a normal signal if $G_t \in \{G^{(c)}\}_{c=1}^{C}$; conversely, the signal is said to be abnormal if it is generated from an outlier graph $G_t \notin \{G^{(c)}\}_{c=1}^{C}$. We define the binary hypothesis classes:

$$(\text{normal}) \ H_0 : G_t \in \{G^{(c)}\}_{c=1}^{C},$$

$$(\text{abnormal}) \ H_1 : G_t \notin \{G^{(c)}\}_{c=1}^{C}. \quad (37)$$

We shall work with cases where $H_1$ occurs with a lower probability than $H_0$ to allow for successful graph inference. Moreover, under $H_1$, the outlier graph is sufficiently different from the normal graphs in terms of its eigencentrality.

As a consequence of Assumption 1, the distance

$$
\min_{c=1,\ldots,C} \|y_t - \Omega_t \circ \mathcal{H}(A^{(c)}_t)B_z\|^2
$$

is expected to be small under $H_0$ and large under $H_1$. While determining such distance would require knowledge of the normal graphs, we utilize the online EM algorithm and replace the latter using up-to-date estimates. This leads to the online detector: let $\kappa > 0$ be a user-defined threshold,

$$
\min_{c=1,\ldots,C} \|y_t - \Omega_t \circ (L^{(c)}_t^{-1} + B^{(-1)}_t)z_t\|^2 \lesssim \kappa \quad (38)
$$

Note that $L^{(c)}_t^{-1} + B^{(-1)}_t \approx \mathcal{H}(A^{(c)}_t)B$ and the estimation quality improves as Algorithm 2 gathers more data samples. We expect the detection performance to improve as $t$ grows. Finally, we incorporate the outlier rejection mechanism into Algorithm 2 by a slight modification; see line 4.

**Remark 3:** Compared to [32], [35], our approach requires additional information on the excitation parameter $z_t$. On the other hand, our approach is capable of simultaneous graph learning and abnormal graph detection.
V. NUMERICAL EXPERIMENTS

In this section, we compare the performance of our EM algorithms on tackling Problem 1 for synthetic and real data with state-of-the-art algorithms.

A. Experiments on Synthetic Data

We generate $C$ core-periphery (CP) graphs with $n = 100$ nodes. For $c = 1, \ldots, C$, the node set $V = \{1, \ldots, n\}$ is partitioned into a core set $V_o^{(c)}$ with size $|V_o^{(c)}| = 10$ and a non-core set $V_p^{(c)} = V \setminus V_o^{(c)}$. Each node in $V_o^{(c)}$ is chosen uniformly at random such that $V_o^{(c)} \neq V_o^{(c')}$ if $c \neq c'$. For any $i, j \in V$, an edge is assigned with probability $1$ if $i, j \in V_o^{(c)}$, with probability $0.2$ if $i \in V_o^{(c)}, j \in V_p^{(c)}$; and with probability $0.05$ if $i, j \in V_p^{(c)}$. Each observed signal $y_t$ is generated through the mixture model (2) with the noise variance of $\sigma^2 = 10^{-2}$. The graph identifier $w_t$ is drawn uniformly from $\{1, \ldots, C\}$. The missing information vector $\Omega_t$ composes of Bernoulli r.v.s with $E[\Omega_{it}] = \gamma \in [0, 1]$. For the excitation, the matrix $B \in \mathbb{R}^{n \times r}$ is generated as $B_{ij} = M_{ij} \bar{B}_{ij}$, where $M_{ij}, \bar{B}_{ij}$ are independent r.v.s, $M_{ij} \in \{0, 1\}$ is Bernoulli with $E[M_{ij}] = 0.1$, and $\bar{B}_{ij} \sim \mathcal{U}(0(0.1, 1))$. The latent parameter matrix $z_t \in \mathbb{R}^r$ is generated as $[z_t]_i = N_{it} \bar{Z}_{it}$, where $N_{it}, \bar{Z}_{it}$ are independent r.v.s, $N_{it} \in \{0, 1\}$ is Bernoulli with $E[N_{it}] = 0.6$, and $\bar{Z}_{it} \sim \mathcal{U}(0(0.1, 1))$. Otherwise, specified exception, the excitation rank will be set as $r = 40$.

We evaluate the performance of (A) central nodes detection and (B) graph signals clustering. For (A), we compute the mismatch between the ground truth $V_o^{(c)}, c = 1, \ldots, C$ and the detected central nodes via the average error rate:

$$\text{Error rate} = 1.0 - (1/C) \sum_c^{C} \mathbb{E} \left[ \frac{1}{10} |V_o^{(c)} \cap \hat{V}_o^{(c)}| \right],$$

(39)

where $\hat{V}_o^{(c)}$ is the top-10 central nodes detected in graph $G^{(c)}$ with the algorithm. For (B), we compute the normalized mutual information (NMI) [63] between the detected graph identifiers $\{\hat{w}_t\}_{t=1}^m$ and the ground truth identifiers $\{w_t\}_{t=1}^m$. A large NMI value indicates a high graph signal clustering accuracy.

Batch Algorithm: We initialize Algorithm 1 by assigning the conditional probabilities using the SC method. Let $U \in \mathbb{R}^{m \times C}$ be the collection of top-$C$ eigenvectors of data correlation matrix $Y^\top Y$ and $\bar{u}_c$ is the centroid vector of the $c$-th cluster computed from $S$. We set:

$$p_c(\Theta^0, DP_t) = \frac{\exp(-||U_{i,c} \cdot \bar{u}_c||^2)}{\sum_c^{C} \exp(-||U_{i,c} \cdot \bar{u}_c||^2)},$$

(40)

We remark that SC gives a good initialization to Algorithm 1 despite that the method alone may not perform well on signals originated from weak low pass filters; see Example 1.

The first experiment considers a batch data setting with $m = 200 C$ signal tuples $\{DP_t\}_{t=1}^m$ generated from (2). The graph filters applied are $H_w(A^{(w_t)}) = (I - \frac{1}{\beta} A^{(w_t)})^{-1}, H_w(A^{(w_t)}) = (I - \frac{1}{\beta} A^{(w_t)})^{-1}$, where $H_w(\cdot)$ is a weaker low pass filter than $H_o(\cdot)$. We set $K_{max} = 100$, $\lambda_L = 0.01$ and $\lambda_S = 0.001$.

We benchmark Algorithm 1 under different probabilities of missing data ($\gamma$) against Graph Laplacian Mixture Model (‘GLMM’) [38], spectral clustering (‘SC’), Spectral Template (‘S. Temp’) [10] and the method by Kalofolias (‘Smooth’) [9]. To infer central nodes using the SC method, we first apply (40) to initialize the E-step and perform only one iteration of the M-step in Algorithm 1 to estimate the low rank and sparse matrices. On the other hand, a three-step procedure is simulated for [9], [10]. We first cluster data into $C$ groups with the graph identifiers from Algorithm 1; then, we apply [9], [10] on the individual data groups to learn the corresponding graphs and compute the eigen-centrality vectors subsequently. To examine the multi-graph assumption, we also take two algorithms based on the single-graph assumption as benchmark: the robust PCA method (‘RPCA’) [19] and the method by Wang (‘MTP’) [43]. We apply them on $m$ samples to detect one group of central nodes $V_o$ and compute the central nodes detection error rate by taking $\hat{V}_o(\cdot) = V_o, c = 1, \ldots, C$.

Fig. 4 compares the performance of algorithms against the number of graphs $C$ with respect to the clustering accuracy (measured by NMI) and centrality detection error rate from 100 Monte-Carlo trials. First, observe that Algorithm 1 achieves significantly better performance than the benchmarks, even when a portion of observations are missing (with $\gamma > 0$). Second, under the weak low pass filter $H_w(\cdot)$ (Fig. 4, Left), the performances of tested algorithms’ worsen with the number of graphs $C$; while the effect of $C$ is less significant with the strong low pass filter $H_o(\cdot)$ (Fig. 4, Right). Our results indicate that Algorithm 1 is robust to smoothness (i.e., low pass property) of graph signals and low-rank excitation.

Online Algorithm: The next experiment considers the streaming data setting where at time $t$, only the $t$th signal tuple $DP_t$ is available and we focus on applying Algorithm 2 to continuously estimate centrality of graphs and cluster the graph signals. The data tuples are generated from (2) in the same way as in the batch data setting with no missing data, i.e., $\gamma = 0$, and we simulate the same pair of strong ($H_o(\cdot)$) and weak ($H_w(\cdot)$) graph filters. Note that we maintain the full dataset with $m = 3000 C$ samples for benchmark. For Algorithm 2, we set $\lambda_L = 0.01, A_S = 0.001$ and initialize the algorithm through applying Algorithm 1 on $m_{init} = 50 C$ signal tuples. The step size is selected as $\beta_t = \frac{0.5}{m_{init}}$. As a benchmark algorithm, we consider apply the batch EM algorithm provided with $\{50 C, 300 C, 900 C, \ldots \}$ samples and apply Algorithm 1 for $K_{max}$ iterations. We also consider a mini-batch setting for Algorithm 2 with the batch size $b_t$.

Fig. 5 plots the trajectories of clustering accuracy through evaluating NMI on the full dataset, centrality detection error rate, and the MAP objective value evaluated over the full dataset with 10 Monte-Carlo runs with the 90% confidence intervals. As observed, the performance of Algorithm 2 improves with time as the algorithm obtains more samples, which allows the algorithm to construct better estimate to the sufficient statistics in (33) via the SA scheme. Comparing between Fig. 5 (Left) and (Right), the terminal performance is affected by the strength of low pass graph filter as well as the model order, i.e., number of candidate graphs. Observe that Algorithm 1 has access to the entire dataset at all iterations and thus gives the best performance. However,
as the iteration number $t$ grows, Algorithm 2 approaches the performance of Algorithm 1.

Lastly, we examine the performance of Algorithm 2 under dynamic graph topology. We consider the same setting as in Fig. 5 with $C = 2$, $H_w(A_{w1})$, but with the exception that after $t = 1500$, the underlying graphs $\{G(c)\}_{c=1}^{C}$ are changed to $\{G_a(c)\}_{c=1}^{C}$. We consider two cases: (a) when 10% of edges are removed, each graph $G_a(c)$ is generated by removing 10% of edges in $G(c)$; (b) when the core nodes are shifted, each graph $G_a(c)$ is generated by replacing one of the core nodes in $V_a^{(c)}$ with one of the periphery nodes in $V_a^{(c)}$, i.e., we first select a core node and a periphery node at random, then we swap all of their neighbours with each other. Fig. 6 shows the graph signal clustering and central nodes detection performances of Algorithm 2 with dynamic graph topology. Observe that the online algorithm accurately tracks the joint inference problem in spite of sudden changes in the graph topology.

\[ \text{Logit model:} \] Before concluding this subsection, let us also consider an application of Algorithm 1 to the logit model with batch data; cf. Section III-A. We consider a set of $C$ graphs built on the simple star graph with $n = 20$ nodes, each with a different central node, and additional edges are assigned with probability 0.02 between the non-central nodes. In each of 30 Monte-Carlo trials, we generate $m = 80C$ data tuples according to (26). The excitation matrix $B$ follows a similar generation process as before but with $E[M_{ij}] = 0.3$. The excitation rank is $r = 16$.
and the tested graph filter is \( \mathcal{H}(A^{(w)}(\cdot)) \). For the logit model, we set the bias parameter \( b \) as the negative average value of all signals. Lastly, Algorithm 1 is implemented in MATLAB with the CVX package [64] for solving (29). Note the benchmark algorithms are implemented through directly treating the binary observations as real-valued graph signals.

Table I compares Algorithm 1 with benchmark algorithms on tackling Problem 1 in terms of the clustering accuracy (NMI) and error rate in detecting the central node of each graph. We observe that Algorithm 1 can accurately separate the observations into \( C \) groups and detect the most central nodes inside C graphs while the benchmarks have failed in almost all 30 trials under the logit model.

### B. Application: Anomaly Detection

This section considers applying Algorithm 2 to online anomaly graph detection application as described in Section IV-A. In the following simulation results, we consider two groups of graphs with the same size \( n = 100 \) and the tested graph filter is \( \mathcal{H}(A^{(w)}(\cdot)) \) from Section V-A. The first group generates normal graph signals from \( C = 2 \) different CP graphs \( \{G^{(c)}\}_{c=1}^{C} \). The second group generates abnormal graph signals via a CP graph \( G^{(C+1)} \) with a different core nodes set \( V_o^{(C+1)} \) from \( \{v_o^{(c)}\}_{c=1}^{C} \). In our simulation, the abnormal graph signals are observed in two modes, either briefly in order or randomly. To initialize Algorithm 2, we use a set of \( m_{\text{init}} \) normal graph signals with the batch Algorithm 1.

We first compare the detector value of (38) against time \( t \) in Fig. 7. The left panel considers the case with \( m_{\text{init}} = 200C \) where Algorithm 2 is initialized with a large batch of normal signals; while the right panel considers the case with \( m_{\text{init}} = 20C \) where the initialization for Algorithm 2 can be inaccurate. With a large batch initialization (left panel), we observe that the detector value (38) records a significant spike over the time intervals with abnormal signals. With small batch initialization (right panel), the detector is less sensitive to the abnormal signals at the beginning. However, as time goes by and Algorithm 2 processes enough samples, (38) produces clear spikes when an abnormal graph signal is recorded.

We next study the change point detection problem which is a special case of anomaly graph detection. Our aim is to detect the time instance when the underlying graph is switched to an abnormal one. For instance, the problem is relevant for detecting events such as transmission line failure in power systems. We consider two types of abnormal graphs: CP graphs with different sets of central nodes as described previously, Barabasi–Albert (BA) graphs where each added node is connected to random \( m_{\text{BA}} = 10 \) existing nodes with probability proportional to their degrees. The first 800 samples are generated from normal graphs, while the remaining 300 samples are generated from the abnormal graph.

We compare the detector values (38) against \( t \) with the culminating sum detector (‘CUSUM’) [32], blind simple matched subspace detector (‘BSMSD’) [35]. Both existing algorithms consider the case with only one normal graph. To extend them into multiple graph settings, for the observed signal \( y_t \) we compute the CUSUM/BSMSD detector values based on each normal graph \( G^{(c)} \) whose topology are assumed known, then we take the minimum of the \( C \) detector values. We expect a spike in the combined detector value at the change point, i.e., the time instance when \( y_t \) is generated from the abnormal graph.

We consider two graph filter designs in our experiment. The first one in Fig. 8 (Left) takes the weak low pass filter \( \mathcal{H}_{\text{w}}(\cdot) \) defined previously. The second one in Fig. 8 (Right) adopts the low pass filter defined in [35]. In the latter case, we set \( B = I \) and the latent parameters are generated as \( z_i \sim N(0, I) \). The eigenvalues of the tested graph filter \( \mathcal{H}_{\text{lb}}(\cdot) \) is \( \Sigma_{\text{lb}}^{(w)}(\cdot) = \exp(-i/10) \), \( i = 1, \ldots, n \).

Fig. 8 presents the detector values against time \( t \) to compare the performance of three tested detectors. We observe that all detectors are able to detect the change point in the setting with graph filter \( \mathcal{H}_{\text{lb}}(\cdot) \), as seen from the pronounced spikes in detector values. Algorithm 2 has a comparable sensitivity to existing works despite the algorithm does not know the graph topology a-priori. On the other hand, under the weak low pass filter \( \mathcal{H}_{\text{w}}(\cdot) \), Algorithm 2 still provides reliable detection on the abnormal signals. The other two detectors do not show any detectable pattern in ‘normal region’ and ‘abnormal region’.

### C. Experiments on Real Data

In this subsection, we apply the proposed algorithms on two datasets of graph signals. The estimated rank of the datasets’ covariance matrix across signals, i.e., \( Y^t Y \) shown in Fig. 9, is the value of \( C \) we applied in the experiment. The first dataset (Stock) is the daily returns of S&P 100 stocks in May 2018 to Aug 2019 with \( n = 98 \) stocks and \( m = 300 \) samples, collected.

### Table I

| Alg | \( S_{\text{Temp}} \) | Smooth | GLMM | SC | RPCA | MTP2 |
|-----|-----------------|--------|------|----|------|------|
| Error rate | 0.15 | 0.73 | 1.00 | 0.71 | 0.91 | 0.58 | 1.00 |
| NMI | 0.93 | 0.18 | 10^{-3} | N/A | N/A | N/A | N/A |
where $Y_{i}:c$ is the $i$th row vector of $Y^{(c)}$, $i = 1, \ldots, n$. Note that the algorithms do not observe the S&P100 index. A higher correlation score indicates the corresponding stock $i$ is a better representative of all stocks as it tracks the market trend given by the S&P100 index, which may imply a more central node. Fig. 10 shows the average correlation scores of top-10 detected central stocks from the corresponding groups of samples clustered by the tested algorithms. Observe that Algorithm 1 delivers higher average correlation scores than other algorithms. We also observe from Table II that Algorithm 1 detects two different groups of central stocks while other algorithms detect pairs of graphs with repeated groups of central stocks. Compared to the detected central stocks under the single-graph assumption, stocks detected by Algorithm 1 under multi-graph assumption have higher correlation scores.

The second dataset (Brain) collects the functional magnetic resonance imaging (fMRI) data of 50 subjects from the Human Connectome Project with $m = 240,000$ samples, where the subjects were in resting state (RS). We use the preprocessed RS fMRI data by [30] and consider the Automated Anatomical Labeling Atlas 90 [65] for labels of brain regions. The $k = 5$ regions with largest summed absolute values of all samples are selected to form the excitation parameters $Z$. They are left & right medial superior frontal gyrus (label 23, 24), left & right cuneus (label 45, 46) and right inferior occipital gyrus (label 54). The remaining $n = 85$ brain regions are regarded as nodes.
in the unknown brain graphs. To initialize Algorithm 2, we take $m_{\text{init}} = 4800$ randomly selected samples and apply the batch algorithm. We set the step size as $\beta_t = \frac{1}{t + \text{max}}$, and the MAP problem parameters are $\lambda_L = 0.45$, $\lambda_S = 0.045$.

The brain connectivity graph can vary for different resting state brain networks [30]. Fig. 11 presents the estimated centrality of $C = 3$ graphs against time while running Algorithm 2. The output of the algorithm stabilizes as more samples are observed, indicating a consistent estimation. Furthermore, the graphs $G^{(2)}$ and $G^{(3)}$ show central nodes that are concentrated in the left brain and right brain, respectively. Table III lists the labels of estimated central brain regions. For $G^{(1)}$, label 70 is Paracentral lobule [66]. It controls the movement and sensation in the lower body. For $G^{(2)}$ and $G^{(3)}$, label 12 and label 13 belong to the inferior frontal gyrus region [67], which is associated with speech and language processing.

VI. CONCLUSION

We study a joint graph inference problem on the challenging mixture model of filtered graph signals under general (non-white) excitation, weak low pass graph filters and missing data. We design an efficient algorithm based on EM and develop the latter’s online extension for streaming data. The online algorithm is further applied to abnormal graph signals detection. Efficacy of the proposed algorithms are verified with convergence analysis and numerical experiments.

APPENDIX A

PROOF OF PROPOSITION 1

The first part follows directly from the steps of EM algorithm as the latter are majorization-minimization iterations for the regularized log-likelihood. For any $k \geq 0$,

$$
\mathcal{L}(\Theta^{k+1}) - \mathcal{L}(\Theta^k) \geq \tilde{\mathcal{L}}(\Theta^{k+1} | \Theta^k) - \tilde{\mathcal{L}}(\Theta^k | \Theta^k) \geq 0, \tag{42}
$$

where the last inequality holds since $\Theta^{k+1}$ is a maximizer to the surrogate $\tilde{\mathcal{L}}(\Theta^{k+1} | \Theta^k)$ (16) and $\tilde{\mathcal{L}}(\Theta^{k+1} | \Theta^k) = L(\Theta^k)$.

For the second part of the proposition, we observe that as

$$
\mathcal{D}(\Theta^k | \Theta^{k-1}) = \mathcal{L}(\Theta^k) - \tilde{\mathcal{L}}(\Theta^k | \Theta^{k-1}) \leq \mathcal{L}(\Theta^k) - \mathcal{L}(\Theta^{k-1}),
$$

where the inequality follows from $\tilde{\mathcal{L}}(\Theta^k | \Theta^{k-1}) \geq \mathcal{L}(\Theta^k | \Theta^{k-1}) = \mathcal{L}(\Theta^{k-1})$. Since the MAP problem has bounded objective value $\mathcal{L}(\Theta) \leq \mathcal{L}^*$. It holds

$$
\sum_{k=1}^{K_{\text{max}}} \mathcal{D}(\Theta^k | \Theta^{k-1}) \leq \mathcal{L}^* - \mathcal{L}(\Theta^0), \tag{43}
$$

for any $K_{\text{max}} \geq 1$. Now, as the gradient w.r.t. $\Theta$ of $\mathcal{D}(\Theta | \Theta)$ is $L$-Lipschitz, it implies

$$
0 \leq \nabla \mathcal{D}(\Theta^k - \frac{1}{t} \nabla \mathcal{D}(\Theta^{k-1} | \Theta^{k-1}) | \Theta^{k-1}) \leq \mathcal{D}(\Theta^k | \Theta^{k-1}) - \frac{1}{2L} \| \nabla \mathcal{D}(\Theta^k | \Theta^{k-1}) \|^2, \tag{44}
$$

$$
\implies \frac{1}{2L} \| \nabla \mathcal{D}(\Theta^k | \Theta^{k-1}) \|^2 \leq \mathcal{D}(\Theta^k | \Theta^{k-1}).
$$
Summing up both sides from $k = 1$ to $k = K_{\text{max}}$ and using (43) yields
\[
\min_{k = 1, \ldots, K_{\text{max}}} \| \nabla D(\Theta^k | \Theta^{k-1}) \|^2 \leq \frac{2L}{K_{\text{max}}} (L^* - L(\Theta^0)).
\] (45)
Furthermore, the directional derivative satisfies
\[
\mathcal{L}'(\Theta^k; \Theta - \Theta^k) = \langle \nabla D(\Theta^k | \Theta^{k-1}) - \Theta - \Theta^k(\Theta^{k-1}) \rangle.
\]
As $\Theta^k$ maximizes the concave function $\widehat{\mathcal{L}}(\Theta | \Theta^{k-1})$, it holds
\[
\widehat{\mathcal{L}}'(\Theta^k; \Theta - \Theta^k | \Theta^{k-1}) \leq 0, \quad \forall \Theta \in \mathcal{T}.
\] (46)
By Cauchy-Schwarz inequality, this implies that
\[
\sup_{\Theta \in \mathcal{T}} \| \mathcal{L}'(\Theta^k; \Theta - \Theta^k) \| \leq \| \nabla D(\Theta^k | \Theta^{k-1}) \|, \quad (47)
\]
with $\frac{\partial}{\partial \Theta} = 0$. Combining with (45) leads to the conclusion.

**APPENDIX B**

**PROOF OF PROPOSITION 2**

To simplify notations in this proof, we denote
\[
\mathcal{L}_{\text{cal}}(\Theta; S) = \Phi(\Theta; S) - \Psi(\Theta),
\]
where $S := (P_c, Y Z_c, Z Z_c, c, c)$ collects the sufficient statistics, and the non-smooth function $\Psi(\Theta)$ is the regularizer and $\Phi(\Theta; S)$ collects the remaining terms as found in (31).

We denote $\mathcal{S}(\Theta)$ as the population sufficient statistics computed from $\Theta$ through (32). Furthermore, $\Theta(\mathcal{S}) \in \arg \max_{\Theta} \mathcal{L}_{\text{cal}}(\Theta; S)$ denotes a maximizer to the surrogate function given the sufficient statistics. Define the smooth function $f(\Theta) := \mathbb{E}_{DP}[ \log p(Y|\Theta, Z, \Omega) ]$ [cf. (35)] and
\[
\Gamma = \{ \Theta : 0 \in \nabla f(\Theta) - \partial \Psi(\Theta) \}.
\] (48)
This is the set of stationary points to the MAP problem (35) with the modified regularizer.

If $\mathcal{S}$ satisfies the fixed point condition (34) where $S = S(\Theta)$, we have $\Theta \in \arg \max_{\Theta} \mathcal{L}_{\text{cal}}(\Theta; S)$ such that
\[
0 \in \partial \mathcal{L}_{\text{cal}}(\Theta; S) = \nabla \Psi(\Theta, S) - \partial \Psi(\Theta).
\] (49)
By the Jensen’s inequality and the fact $f(\Theta) - f(\Theta) = \Phi(\Theta, S) - \Phi(\Theta, \Theta, S) \geq c, \forall \Theta$, (50) and the lower bound is achieved when $\Theta = \overline{\Theta}$. It implies
\[
0 = \nabla f(\overline{\Theta}) - \nabla \Psi(\overline{\Theta}, S).
\] (51)
Inserting (51) into the fixed point condition (49) yields $\overline{\Theta} \in \Gamma$.
On the other hand, let $\overline{\Theta} \in \Gamma$ and observe the inequality (50) and its derived condition (51). Thus,
\[
0 \in \nabla \Psi(\overline{\Theta}; S) - \partial \Psi(\overline{\Theta}),
\] (52)
As the maximizer to $\mathcal{L}_{\text{cal}}(\Theta; S)$ is unique for any $\Theta$, $\Theta(\mathcal{S})$ is well defined and we conclude that $\overline{\Theta} = \Theta(\mathcal{S})$. Furthermore, it holds that $\mathcal{S} = S(\overline{\Theta})$, which is a fixed point to (34).
S. Shaked and T. Routtenberg, “Identification of edge disconnections in graph signals,” IEEE Trans. Signal Process., vol. 69, pp. 314–327, 2021.

S. Ray, A. Buciulea, M. Navarro, S. A. Marques, C. Uhler, and S. Segarra, “Joint inference of multiple graphs from matrix polynomials,” J. Mach. Learn. Res., vol. 23, pp. 1–35, 2020.

X. Yang, M. Sheng, Y. Yuan, and T. Q. S. Quek, “Network topology inference from heterogeneous incomplete graph signals,” IEEE Trans. Signal Process., vol. 69, pp. 5817–5821, 2021.

I. Ricchi, A. Tarun, H. P. Maretic, P. Frossard, and D. Van De Ville, “Dynamics of functional network organization through graph mixture learning,” Neuroimage, vol. 252, 2022, Art. no. 119037.

R. H. Heiberger, “Stock network stability in times of crisis,” Physica A: Stat. Mech. Appl., vol. 393, pp. 376–381, 2014.

C. Kaushik, T. M. Roddenberry, and S. Segarra, “Network topology change-point detection from graph signals with prior spectral signatures,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2021, pp. 5395–5399.

S. P. Chepuri and G. Leus, “Subgraph detection using graph signals,” in Proc. IEEE 50th Asilomar Conf. Signals, Syst. and Comput., 2016, pp. 532–536.

S. Shaked and T. Rountenborg, “Identification of edge disconnections in networks based on graph filter outputs,” IEEE Trans. Signal Inf. Process. Over Netw., vol. 7, pp. 578–594, 2021.

E. Isufi, A. S. Mahabir, and G. Leus, “Blind graph topology change detection,” IEEE Signal Process. Lett., vol. 25, no. 5, pp. 655–659, May 2018.

U. V. Luxburg, “A tutorial on spectral clustering,” Statist. Comput., vol. 17, no. 4, pp. 435–466, 2007.

G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “KNN model-based approach in classification,” in Proc. ODBASE, 2003, pp. 986–996.

H. P. Maretic and P. Frossard, “Graph Laplacian mixture model,” IEEE Trans. Signal Inf. Process. Over Netw., vol. 6, pp. 261–270, 2020.

J. Yuan, D. Wang, N. Soh, and X. Yang, K. Guo, and T. Q. S. Quek, “GRACGE: Graph signal clustering and multiple graph estimation,” IEEE Trans. Signal Process., vol. 70, pp. 2015–2030, 2022.

A. Karasaanli and S. Aviyente, “Simultaneous graph signal clustering and graph learning,” in Proc. Int. Conf. Mach. Learn., 2022, pp. 10762–10772.

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.

S. Kumar, J. Ying, J. V. de Miranda Cardoso, and D. P. Palomar, “A unified framework for structured graph learning via spectral constraints,” J. Mach. Learn. Res., vol. 21, no. 22, pp. 1–60, 2020.

Y. Wang, U. Roy, and C. Uhler, “Learning high-dimensional gaussian graphical models under total positivity without adjustment of tuning parameters,” in Proc. Int. Conf. Artif. Intell. Statist., 2020, pp. 2698–2708.

J. V. d. M. Cardoso, J. Ying, and D. P. Palomar, “Algorithms for learning graphs in financial markets,” 2020, arXiv:2012.15410.

E. Pircalabelu and G. Claeskens, “Community-based group graphical lasso,” J. Mach. Learn. Res., vol. 13, no. 1, pp. 2406–2434, 2020.

S. Vlaski, H. P. Maretic, R. Nassif, P. Frossard, and A. H. Sayed, “Online graph learning from sequential data,” in Proc. IEEE Data Sci. Workshop, 2018, pp. 190–194.

R. Shahpour and G. Mateos, “Online topology inference from streaming stationary graph signals with partial connectivity information,” Algorithms, vol. 13, no. 9, 2020, Art. no. 228.

V. Shanovskaia, K. Ntunob, S. Vlaski, and A. H. Sayed, “Online graph learning from social interactions,” in Proc. IEEE 55th Asilomar Conf. Signals, Systems, Comput., 2021, pp. 1263–1267.

S. S. Saboksary, G. Mateos, and M. Cetin, “Online discriminative graph learning from multi-class smooth signals,” Signal Process., vol. 186, 2022, Art. no. 108101.

M. Udell and A. Townsend, “Why are Big Data matrices approximately low rank?,” SIAM J. Math. Data Sci., vol. 1, no. 1, pp. 144–160, 2019.

M. Cucuringu, P. Rombach, S. H. Lee, and M. A. Porter, “Detection of core–periphery structure in networks using spectral methods and geodesic paths,” Eur. J. Appl. Math., vol. 27, no. 6, pp. 846–887, 2016.

R. Ramakrishna, H.-T. Wai, and A. Scaglione, “A user guide to low-pass graph signal processing and its applications,” IEEE Signal Process. Mag., vol. 37, no. 6, pp. 74–85, Nov. 2020.

A. Agarwal et al., “Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions,” Ann. Statist., vol. 40, no. 2, pp. 1171–1192, 2012.

J. Mairal, “Incremental majorization-minimization optimization with application to large-scale machine learning,” SIAM J. Optim., vol. 25, no. 2, pp. 829–855, 2015.

B. Karimi, H.-T. Wai, E. Moulines, and P. Li, “Minimization by incremental stochastic surrogate optimization for large scale nonconvex problems,” in Proc. 33rd Int. Conf. Algorithmic Learning Theory, vol. 167, 2022, pp. 606–637.

A. Aravkin, S. Becker, V. Cevher, and P. Olsen, “A variational approach to stable principal component pursuit,” in Proc. 30th Conf. Uncertainty Artif. Intell., 2014, pp. 32–41.

H. Robbins and S. Monro, “A stochastic approximation method,” The Ann. Math. Statist., vol. 22, pp. 400–407, 1951.

O. Cappé and E. Moulines, “On-line expectation-maximization algorithm for latent data models,” J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.), vol. 71, no. 3, pp. 593–613, 2009.

B. Karimi, B. Miaoosjedow, E. Moulines, and H.-T. Wai, “Non-asymptotic analysis of biased stochastic approximation scheme,” in Proc. Conf. Learn. Theory, 2019, pp. 1944–1974.

H. D. Nguyen, F. Forbes, G. Fort, and O. Cappé, “An online minimization-maximization algorithm,” in Proc. 17th Conf. Int. Federation Classification Soc., Porto, Portugal, Jul. 2022.

H. Kushner and G. G. Yin, Stochastic Approximation and Recursive Algorithms and Applications. Berlin, Germany: Springer, 2003.

S. Balakrishnan, M. J. Wainwright, and B. Yu, “Statistical guarantees for the EM algorithm: From population to sample-based analysis,” Ann. Statist., vol. 45, no. 1, pp. 77–120, 2017.

N. X. Vinh, I. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance,” J. Mach. Learn. Res., vol. 11, pp. 2837–2854, 2010.

M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” 2014, [Online], Available: http://cvxr.com/cvx

N. Tzourio-Mazoyer et al., “Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain,” Neuroimage, vol. 15, no. 1, pp. 273–289, 2002.

J. C. Grotta et al., Stroke E-Book: Pathophysiology, Diagnosis, and Management. New York, NY, USA: Elsevier, 2021.

J. D. Greenlee et al., “Functional connections within the human inferior frontal gyrus,” J. Comp. Neurol., vol. 503, no. 4, pp. 550–559, 2007.