

Temporal Multiresolution Graph Learning

KOKI YAMADA¹, (Member, IEEE), AND YUICHI TANAKA^{1,2}, (Senior Member, IEEE)

¹Department of EECS, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-8588, Japan

²PRESTO, Japan Science and Technology Agency (JST), Kawaguchi 332-0012, Japan

Corresponding author: Koki Yamada (k-yamada@msp-lab.org)

This work was supported in part by the Japan Science and Technology Agency (JST) Precursory Research for Embryonic Science and Technology (PRESTO) under Grant JPMJPR1935, and in part by Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 20H02145 and Grant 20J13647.

ABSTRACT Estimating time-varying graphs, i.e., a set of graphs in which one graph represents the relationship among nodes in a certain time slot, from observed data is a crucial problem in signal processing, machine learning, and data mining. Although many existing methods only estimate graphs with a single temporal resolution, the actual graphs often demonstrate different relationships in different temporal resolutions. In this study, we propose an approach for time-varying graph learning by leveraging a multiresolution property. The proposed method assumes that time-varying graphs can be decomposed by a linear combination of graphs localized at different temporal resolutions. We formulate a convex optimization problem for temporal multiresolution graph learning. In experiments using synthetic and real data, the proposed method demonstrates the promising objective performances for synthetic data, and obtains reasonable temporal multiresolution graphs from real data.

INDEX TERMS Graph inference, graph learning, time-varying graph, temporal multiresolution graph.

I. INTRODUCTION

Many applications of signal processing, machine learning, and data mining require the handling of sensor data, where the sensors are often distributed nonuniformly in a physical space. Analyzing such data by considering their underlying spatial structure, i.e., network, can significantly improve the quality of data analysis. Graphs are useful tools to mathematically represent such networks.

Over the last decade, considerable research has been conducted on processing signals on graphs, i.e., *graph signal processing* (GSP) [1]–[4]. GSP has found many applications such as brain signal analysis [5], [6], image processing [7]–[9], and sensor networks [10].

In many cases, graphs are not given a priori. Therefore, *graph learning* [11]–[14], techniques and algorithms for estimating a graph from observed data and/or feature values, are required in various GSP applications, particularly for sensor measurements.

Various graph learning methods have been proposed thus far, which are summarized in two overview papers [13], [15]. Most of these methods belong to a class of *static graph learning* (SGL), which represents the task of learning a single graph from a set of available data. SGL assumes that all the

data follow the same signal generation model. However, there often exist time-varying relationships in many applications. In other words, underlying networks vary over time wherein the observed data follow a time-varying generation model. An example of a time-varying generation models is the brain functional connectivity from EEG or fMRI data, where signals can be thought of as generated by evolving brain networks. Furthermore, we can easily identify the model from the seasonal behavior of sensor measurements.

To handle these dynamic behaviors, *time-varying graph learning* (TVGL) methods have been proposed [12], [16]–[18]. Technically, a time-varying graph consists of multiple graphs where one graph corresponds to the relationship among vertices in a certain time slot. Typically, TVGL divides multivariate time-series data into consecutive (overlapped or nonoverlapped) data segments, and learns multiple graphs from these segments.

TVGL demonstrates a trade-off in the temporal resolution. For example, a large time window allows the capture of a global structure but results in the loss of the local temporal behavior. In contrast, selecting a small time window enables the capture of fast-changing behaviors but may also result in noise sensitivity. To tackle this problem, most existing TVGL methods [12], [16], [17] impose constraints for temporal variations of graphs between neighboring time slots. However, they have two main limitations. First, they are not

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino¹.

suitable when the data are not fitted to the prior assumption of temporal variations. Second, even if the temporal variation information is known, it is often difficult to determine the hyperparameter(s) such as those in the constraints on the temporal variation and the temporal window size.

In this study, we propose a TVGL method without using a specific prior for network evolution. Instead, we assume that the time-varying graphs have a *temporal multiresolution (TMR) structure*: They can be represented by the combination of graphs at different temporal resolutions from local (i.e., short time period) to global (i.e., static) ones. This is desirable because multivariate time-series data tend to have a multiresolution property. An example of such TMR data is temperature data observed in multiple sensor locations. The measurements in each location exhibit the interdependence relationships which change hourly, daily, monthly, and even yearly, where the relationships correspond to structures localized at different temporal resolutions. Our proposed method automatically reveals which edge sets are localized at which temporal variation. In our problem setting, temporal resolution levels correspond to time window sizes in the time-varying graphs, but are not necessarily set as a hyperparameter.

The proposed TVGL is formulated as a convex optimization problem derived from the generation model. We also present an iterative algorithm for solving the optimization problem efficiently, which guarantees the convergence of the solution.

In experiments with synthetic datasets, the proposed method demonstrates superior performance to that of conventional single-resolution TVGL methods. Experiments on a real climate dataset demonstrate that the proposed methods can learn reasonable time-varying graphs that capture seasonal and geographical characteristics, which experimentally proves our concept.

Our preliminary work was presented in [19]. The present study significantly extends our previously proposed approach to simultaneously learn graphs localized at multiple temporal resolutions.

The remainder of this paper is organized as follows. Related works on the proposed methods are summarized in Section II. Notations used in this paper and preliminaries are defined in Section III. We present a generic TVGL framework from multivariate time-series signals in Section IV. Section V presents the proposed TMR graph learning method. Experimental results with synthetic and real data are presented in Section VI. Finally, we present our conclusions in Section VII.

II. RELATED WORKS

Many methods for graph learning have been proposed thus far. Most of them are summarized in the two overview papers [13], [15]. Without being exhaustive, we review SGL and TVGL methods related to our approach.

The basic strategy of SGL is the design of optimization problems based on some desired criteria for learned

graphs. For example, [20]–[22] assume that signals are smooth on a graph. This characteristic is often represented as the Laplacian quadratic form. Instead of smoothness, [11], [23], [24] assume that signals are generated from a Laplacian constrained Gaussian Markov random field (LGMRF), and maximize its regularized likelihood. Some studies such as like [11], [13], [15] suggest a relation between the signal smoothness and the LGMRF likelihood; the smoothness-based approach in [21] solves a relaxed problem of the LGMRF likelihood criterion.

As TVGL methods, [12], [16], [17] learn time-varying graphs with the assumption of signal smoothness and impose constraints on the network evolution. These TVGL methods are designed to learn time-varying graphs with a user-specified single temporal resolution (i.e., window size).

In a different line of GSP research, some graph learning methods assume that observations are generated by applying some filters, e.g., graph variation and heat diffusion operators, to a latent signal [25]–[27]. Their extensions to TVGL, proposed in [28], [29], focus on estimating graphs and corresponding filters simultaneously under the assumption on stationarities of graph signals or signal generation models based on graph filters. Such a simultaneous estimation is out of the scope of this study. The method proposed in this study, in contrast, is based on a different signal generation model. Note that all of the previous works mentioned above are single temporal resolution TVGL approaches.

In contrast to the existing approaches, we estimate graphs having multiple temporal resolutions to capture various temporal relationships. To the best of our knowledge, this is the first attempt in which TVGL has been used to extract a TMR behavior.

Some studies focus on learning multiple graphs (not necessarily time-varying) from observations. While they yield multiple graphs, the learned graphs may not represent time-varying relationships [30], [31].

From a machine learning perspective, TVGL relates to time-varying inverse covariance estimation [32]–[35]. The main difference between TVGL and inverse covariance estimation is whether the optimization problem contains the constraint on the graph Laplacian. For example, a well-known inverse covariance estimation, graphical Lasso [36], yields a covariance matrix that corresponds to a graph with negative edges and self-loops. This would be inappropriate if we need to learn time-varying graphs with nonnegative edge weights without self-loops, which is a typical assumption of GSP. In contrast, our approach constrains a solution space for graphs such that they have nonnegative edges and no self-loops.

III. NOTATION AND PRELIMINARIES

A. NOTATION

Lowercase normal, lowercase bold, and uppercase bold letters denote scalars, vectors, and matrices, respectively. Both X_{ij} and $[\mathbf{X}]_{i,j}$ represent the (i, j) entry of the matrix \mathbf{X} . Furthermore, the j th column vector of \mathbf{X} is denoted by $[\mathbf{X}]_j$. 1

is an all-one vector. $\mathbf{X} \circ \mathbf{Y}$ represents the Hadamard product of \mathbf{X} and \mathbf{Y} . The Moore–Penrose pseudo inverse of \mathbf{X} is denoted by \mathbf{X}^\dagger .

B. BASIC DEFINITIONS FOR GRAPHS

A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ is a graph with a vertex set \mathcal{V} and an edge set \mathcal{E} , where the number of nodes and edges are denoted by $N = |\mathcal{V}|$ and $E = |\mathcal{E}|$, respectively. $\mathbf{W} \in \mathbb{R}^{N \times N}$ denotes a weighted adjacency matrix whose (i, j) -th element represents an edge from the i th vertex to the j th vertex. In this study, we assume that \mathcal{G} is undirected with nonnegative edge weights and does not have self-loops, i.e., \mathbf{W} is symmetric with nonnegative elements and all its diagonal elements are zero. Graph Laplacian is given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where \mathbf{D} is the degree matrix defined as $D_{ii} = \sum_j W_{ij}$.

C. GRAPH LAPLACIAN OPERATOR

Let $\mathbf{w} \in \mathbb{R}_+^{N(N-1)/2}$ be a vector composed of edge weights. It corresponds to the vectorized version of the lower triangular part in \mathbf{W} (excluding diagonal elements). The graph Laplacian operator $\mathcal{L} : \mathbf{w} \in \mathbb{R}_+^{N(N-1)/2} \rightarrow \mathcal{L}\mathbf{w} \in \mathbb{R}^{N \times N}$ is defined as [24]

$$[\mathcal{L}\mathbf{w}]_{ij} = \begin{cases} -w_{i+d_j} & i > j \\ [\mathcal{L}\mathbf{w}]_{j,i} & i < j \\ \sum_{i \neq j} [\mathcal{L}\mathbf{w}]_{i,j} & i = j \end{cases} \quad (1)$$

where $d_j = -j + \frac{j-1}{2}(2N-j)$. Simply speaking, \mathcal{L} transforms \mathbf{w} into \mathbf{L} . Its adjoint operator $\mathcal{L}^* : \mathbf{Y} \in \mathbb{R}^{N \times N} \rightarrow \mathcal{L}^*\mathbf{Y} \in \mathbb{R}_+^{N(N-1)/2}$ is given by

$$[\mathcal{L}^*\mathbf{Y}]_k = y_{ii} - y_{ij} - y_{ji} + y_{jj}, \quad k = i - j + \frac{j-1}{2}(2N-j), \quad (i > j). \quad (2)$$

D. PRIMAL-DUAL SPLITTING ALGORITHM

Let $\Gamma(\mathbb{R}^n)$ be the set of all appropriate lower semicontinuous convex functions on \mathbb{R}^n . A primal-dual splitting algorithm [37] can solve the following optimization problem for some $M \in \mathbb{N}$:

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) + \sum_{i=1}^M h_i(\mathcal{A}_i\mathbf{w}), \quad (3)$$

where $f \in \Gamma(\mathbb{R}^n)$ is a ξ -Lipschitz differentiable convex function with $\xi > 0$, $g \in \Gamma(\mathbb{R}^n)$ and $h_i \in \Gamma(\mathbb{R}^m)$ are functions that a proximal operator can be computed efficiently, and \mathcal{A}_i is a linear operator. The proximal operator $\text{prox}_{\gamma f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ of f with a parameter $\gamma > 0$ is defined by

$$\text{prox}_{\gamma f}(\mathbf{x}) = \arg \min_{\mathbf{y}} f(\mathbf{y}) + \frac{1}{2\gamma} \|\mathbf{y} - \mathbf{x}\|_2^2. \quad (4)$$

If f is separable across variables such as $f(\mathbf{x}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$ with $\mathbf{x} = [\mathbf{x}_1^\top, \mathbf{x}_2^\top]^\top$, its proximal operator can be reduced to the proximal operator for each of them as follows:

$$\text{prox}_{\gamma f}(\mathbf{v}) = [(\text{prox}_{\gamma f_1}(\mathbf{v}_1))^\top, (\text{prox}_{\gamma f_2}(\mathbf{v}_2))^\top]^\top, \quad (5)$$

where $\mathbf{v} = [\mathbf{v}_1^\top, \mathbf{v}_2^\top]^\top$.

Let $\gamma_1, \gamma_2 > 0$ be parameters satisfying the convergence condition $\gamma_1(\frac{\xi}{2} + \gamma_2 \|\sum_{i=1}^M \mathcal{A}_i^* \mathcal{A}_i\|) < 1$. The primal-dual splitting algorithm is given by the following iteration:

$$\begin{cases} \mathbf{x}^{(n+1)} := \text{prox}_{\gamma_1 g}(\mathbf{x}^{(n)} - \gamma_1(\nabla f(\mathbf{x}^{(n)}) + \sum_i \mathcal{A}_i^* \mathbf{y}^{(n)})) \\ \text{For } i = 1, \dots, M \\ \mathbf{y}_i^{(n+1)} := \text{prox}_{\gamma_2 h_i^*}(\mathbf{y}_i^{(n)} + \gamma_2 \mathcal{A}_i(2\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)})) \end{cases} \quad (6)$$

where h_i^* in the last equation is the convex conjugate function of h_i .

IV. LEARNING GRAPHS WITH LGMRF

Here, we present an overview of a generic formulation of SGL and (single-resolution) TVGL methods based on LGMRF in the literature because our formulation also leverages it. The formulation is also useful to distinguish the proposed method from existing methods.

Several existing approaches can be written using this generic formulation, although they have been proposed independently. We derive some representative graph learning methods from the generic formulation and reveal the relationship between them.

A. GENERAL FORMULATION

Suppose that multivariate time-series data $\{\mathbf{x}_t\}_{t=0}^{T-1}$ are given where $\mathbf{x}_t \in \mathbb{R}^N$ and T is the duration of the time-series. We also assume that \mathbf{x}_t is generated by LGMRF as follows:

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{L}_t) &= (2\pi)^{-N/2} \left(\text{gdet}(\mathbf{L}_t^\dagger) \right)^{-1/2} \exp \left(-\frac{1}{2} \mathbf{x}_t^\top \mathbf{L}_t \mathbf{x}_t \right), \quad (7) \end{aligned}$$

where \mathbf{L}_t is the graph Laplacian at time t that corresponds to the underlying graph of \mathbf{x}_t , and gdet represents the generalized determinant [38]. Letting $\mathbf{L}_t = \mathcal{L}\mathbf{w}_t$, $\mathbf{w}_t \in \mathbb{R}^E$, and $E = N(N-1)/2$, (7) can be rewritten as

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{w}_t) &= (2\pi)^{-N/2} \left(\text{gdet}((\mathcal{L}\mathbf{w}_t)^\dagger) \right)^{-1/2} \\ &\quad \times \exp \left(-\frac{1}{2} \mathbf{x}_t^\top \mathcal{L}\mathbf{w}_t \mathbf{x}_t \right). \quad (8) \end{aligned}$$

The edge weight vector \mathbf{w}_t will be sparse and nonnegative. Assume that the prior distribution of \mathbf{w}_t is the following E -variate exponential distribution [11]:

$$p(\mathbf{w}_t) = \left(\frac{\alpha}{2} \right)^E \exp \left(-\frac{\alpha}{2} \mathbf{1}^\top \mathbf{w}_t \right) \quad \text{for } \mathbf{w}_t \geq \mathbf{0}, \quad (9)$$

The maximum a posteriori (MAP) estimation of $p(\mathbf{x}_t | \mathbf{w}_t)$ leads to the following optimization problem:

$$\min_{\mathbf{w}_0 \dots \mathbf{w}_{T-1} \geq \mathbf{0}} \sum_{t=0}^{T-1} -\log \text{gdet}(\mathcal{L}\mathbf{w}_t) + \alpha \|\mathbf{w}_t\|_1 + \mathbf{x}_t^\top \mathcal{L}\mathbf{w}_t \mathbf{x}_t, \quad (10)$$

where α controls the sparsity of the graph.

The optimization problem in (10) is the general form for the SGL and TVGL problems based on LGMRF. We also

utilize (10) for the multiresolution TVGL proposed in this study.

While the problem itself is generic, directly solving this problem is impractical because it needs to learn one graph from one data sample due to overfitting. Therefore, we often need to divide the data into multiple segments. By assuming that the data within the same segment have one common graph, (10) is reduced to well-known SGL and TVGL problems. In the following subsections, we derive representative SGL and TVGL settings from (10).

B. STATIC GRAPH LEARNING

Suppose that \mathbf{w}_t is constant for all time instances t , i.e., the graph is static over time. This leads to $\mathbf{w}_t = \mathbf{w}$, $t = 0, \dots, T - 1$. Then, (10) is reduced to the following problem:

$$\min_{\mathbf{w} \geq 0} -\log \text{gdet}(\mathcal{L}\mathbf{w}) + \alpha \|\mathbf{w}\|_1 + \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{x}_t^\top \mathcal{L}\mathbf{w}\mathbf{x}_t. \quad (11)$$

The third term in (11) represents the smoothness of the signal as the Laplacian quadratic form $\mathbf{x}_t^\top \mathcal{L}\mathbf{x}_t$. This term can be rewritten by using the sample covariance matrix \mathbf{S} as

$$\sum_{t=0}^{T-1} \mathbf{x}_t^\top \mathcal{L}\mathbf{w}\mathbf{x}_t = \text{tr}\left(\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{x}_t^\top \mathcal{L}\mathbf{w}\right) = T \text{tr}(\mathbf{S}\mathcal{L}\mathbf{w}). \quad (12)$$

As a result, (11) can be rewritten using (12) as

$$\min_{\mathbf{w} \geq 0} -\log \text{gdet}(\mathcal{L}\mathbf{w}) + \alpha \|\mathbf{w}\|_1 + \text{tr}(\mathbf{S}\mathcal{L}\mathbf{w}). \quad (13)$$

It is equivalent to the graphical Lasso problem with graph Laplacian constraints [11].

C. TIME-VARYING GRAPH LEARNING

As previously mentioned, learning one graph from one sample from (10) causes overfitting. Instead, we consider a TVGL problem by dividing the time-series data with nonoverlapping time windows in the same manner as [12], [16], [17].

Let $\mathbf{X}^{(k)} = [\mathbf{x}_{kr}, \dots, \mathbf{x}_{(k+1)r-1}]$ ($k = 0, \dots, K - 1$) be the k th data chunk where r is the time window size and k is the index for the time window. We denote $\mathbf{w}^{(k)}$ as the edge weight vector corresponding to the underlying graph of $\mathbf{X}^{(k)}$.

Under the assumption that the graph within the same time window is fixed, i.e., $\mathbf{w}_t = \mathbf{w}^{(k)}$, ($t = kr, \dots, (k + 1)r - 1$), TVGL is formulated as follows:

$$\min_{\mathbf{w}^{(0)} \dots \mathbf{w}^{(K-1)} \geq 0} \sum_{k=0}^{K-1} -\log \text{gdet}(\mathcal{L}\mathbf{w}^{(k)}) + \alpha \|\mathbf{w}^{(k)}\|_1 + \frac{1}{r} \text{tr}((\mathbf{X}^{(k)})^\top \mathcal{L}\mathbf{w}^{(k)} \mathbf{X}^{(k)}) + \beta \sum_{k=1}^{K-1} \psi(\mathbf{w}^{(k)} - \mathbf{w}^{(k-1)}) \quad (14)$$

where $\psi(\cdot)$ is an additional regularizer that characterizes the temporal evolution based on the prior knowledge of time-varying graphs and β is its parameter. Note that the problem is identical to the SGL in (11) if $\psi(\cdot) = 0$.

As possible regularizers, $\psi(\cdot) = \|\cdot\|_2^2$ reflects a time-varying graph whose edge weights change smoothly

over time, and $\psi(\cdot) = \|\cdot\|_1$ leads to the graph wherein only a small number of edges change at any given time. A similar problem to (14) is proposed in [18].

This approach is effective as long as we have appropriate prior knowledge of the temporal evolution, i.e., $\psi(\cdot)$, and the accurate window size r . However, an inappropriate choice of $\psi(\cdot)$ or r leads to inappropriate graphs. To tackle this problem, we propose TMR TVGL in the next section.

V. MULTIREOLUTION TIME-VARYING GRAPH LEARNING

In this section, we present the formulation of the TMR TVGL and an algorithm for solving it.

A. FORMULATION

Here, we introduce a TVGL method that learns $\{\mathbf{w}_t\}_{t=0}^{T-1}$ based on a multiresolution assumption. For simplicity, suppose that T is divisible by 2^L , however, this method is applicable to general values of T .

Suppose that \mathbf{W}_t can be represented as a combination of graphs localized at a temporal resolution $\mathbf{W}_{l,m}$, as illustrated in Fig. 1. We refer to $\mathbf{W}_{l,m}$ as the TMR graph at the temporal resolution l and the segment index m . Therefore, the multi-scale representation of $\{\mathbf{w}_t\}_{t=0}^{T-1}$ is given by the sum of TMR graphs corresponding to time t as

$$\mathbf{w}_t \approx \bar{\mathbf{w}}_t = \sum_{i=0}^L \mathbf{w}_{i, \lfloor q(t)/2^{L-i} \rfloor}, \quad (15)$$

where $q(t) = \lfloor \frac{t}{T} 2^L \rfloor$ and L is the maximum temporal resolution level.

This TMR representation has two advantages. First, it reduces the number of parameters to learn. For TMR

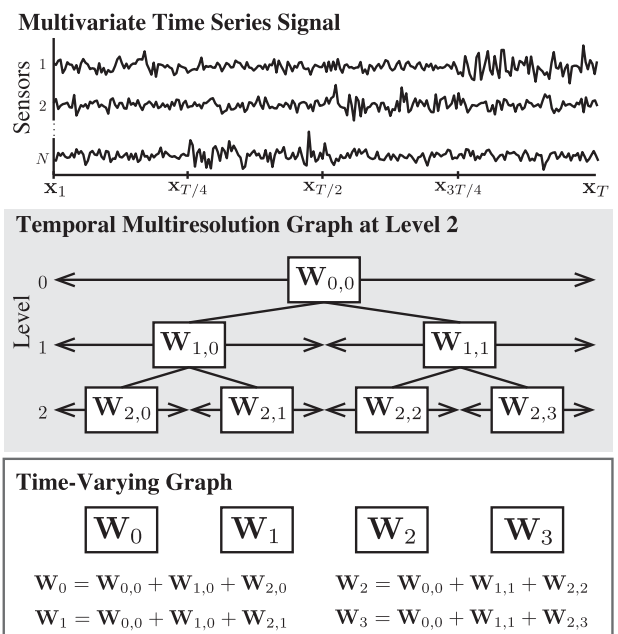


FIGURE 1. Overview of multiresolution graph learning.

TVGL, we need $E(2^{(L+1)} - 1)$ parameters, whereas the number of parameters in a single-resolution TVGL is ET . $E(2^{(L+1)} - 1) \leq ET$ when $L \leq \log_2 T - 1$. It is beneficial if we only have a limited amount of available data. Second, the TMR representation enables the capture of the edges localized in an arbitrary temporal resolution, without specifying the temporal window size.

Now, we consider the detailed formulation of the proposed TVGL. The goal is to learn $\mathbf{w}_{l,m}$ from $\{\mathbf{x}_t\}_{t=0}^{T-1}$. Substituting (15) into (10) leads to the following problem:

$$\min_{\mathbf{w}_{0,0}, \dots, \mathbf{w}_{L,2^L-1} \geq 0} \sum_{t=0}^{T-1} -\log \text{gdet} \left(\mathcal{L} \left(\sum_{k=0}^L \mathbf{w}_{k, \lfloor q(t)/2^{(L-k)}} \right) \right) + \alpha \left\| \sum_{k=0}^L \mathbf{w}_{k, \lfloor q(t)/2^{(L-k)}} \right\|_1 + \sum_{k=0}^L \mathbf{x}_t^\top \mathcal{L} \left(\sum_{k=0}^L \mathbf{w}_{k, \lfloor q(t)/2^{(L-k)}} \right) \mathbf{x}_t. \quad (16)$$

Letting

$$\mathbf{F} = [\mathbf{w}_{0,0}, \mathbf{w}_{1,0}, \mathbf{w}_{1,1}, \dots, \mathbf{w}_{L,2^L-1}] \in \mathbb{R}^{E \times 2^{(L+1)-1}},$$

$$\mathbf{X}_k = [\mathbf{x}_{kr}, \dots, \mathbf{x}_{kR+R-1}] \quad (k = 0, \dots, 2^L - 1, R = T/2^L), \quad (17)$$

(16) can be rewritten as:

$$\min_{\mathbf{F} \geq 0} \alpha \|\mathbf{F}\mathbf{M}\|_1 + \sum_{k=0}^{2^L-1} \frac{1}{R} \text{tr}((\mathbf{X}_k)^\top \mathcal{L}([\mathbf{F}\mathbf{M}]_k) \mathbf{X}_k) - \log \text{gdet}(\mathcal{L}([\mathbf{F}\mathbf{M}]_k)), \quad (18)$$

where $\mathbf{M} \in \mathbb{R}^{2^{(L+1)-1} \times 2^L}$ is given by

$$[\mathbf{M}]_{i,j} = \begin{cases} 1 & 2^{L-l}m \leq j \leq 2^{L-l}(m+1) - 1 \\ 0 & \text{otherwise,} \end{cases}$$

in which $l = \lfloor \log_2(i+1) \rfloor$ and $m = \text{mod}(i+1, 2^l)$. Note that $[\mathbf{F}\mathbf{M}]_k = \bar{\mathbf{w}}_{kR} = \dots = \bar{\mathbf{w}}_{kR+R-1}$.

In (18), we need to obtain a sparse $\mathbf{w}_{l,m}$ to capture the temporally localized structure. However, the direct constraint on the sparseness of $\mathbf{F}\mathbf{M}$ does not result in a sparse \mathbf{F} . Therefore, we replace the first term in (18) with the sparse constraint on \mathbf{F} as follows:

$$\min_{\mathbf{F} \geq 0} \alpha \|\mathbf{F}\|_1 + \sum_{k=0}^{2^L-1} \frac{1}{R} \text{tr}((\mathbf{X}_k)^\top \mathcal{L}([\mathbf{F}\mathbf{M}]_k) \mathbf{X}_k) - \log \text{gdet}(\mathcal{L}([\mathbf{F}\mathbf{M}]_k)). \quad (19)$$

This is the proposed TVGL formulation for learning TMR graphs. In the following subsection, we describe an algorithm to solve (19).

B. ALGORITHM

The optimization problem in (19) is convex and can be solved using the PDS algorithm. Here, we reformulate (19) to the PDS applicable form.

Let $\mathbf{Z}_k \in \mathbb{R}^{N \times N}$ be a pairwise distance matrix computed from

$$[\mathbf{Z}_k]_{i,j} = \sum_{n=0}^{R-1} \|[\mathbf{X}_k]_{i,n} - [\mathbf{X}_k]_{j,n}\|^2, \quad (20)$$

and $\mathbf{z}_k \in \mathbb{R}^E$ be the vector form representation of \mathbf{Z}_k . The third term of (19) can then be rewritten as

$$\sum_{k=0}^{2^L-1} \text{tr}((\mathbf{X}_k)^\top \mathcal{L}([\mathbf{F}\mathbf{M}]_k) \mathbf{X}_k) = \sum_{k=0}^{2^L-1} \mathbf{z}_k^\top [\mathbf{F}\mathbf{M}]_k = \|\mathbf{Z}_{\text{all}} \circ (\mathbf{F}\mathbf{M})\|_1 = \|(\mathbf{M}\mathbf{Z}_{\text{all}}^\top) \circ \mathbf{F}^\top\|_1 \quad (21)$$

where $\mathbf{Z}_{\text{all}} = [\mathbf{z}_0, \dots, \mathbf{z}_{2^L-1}]$. Here, we denote $\bar{\mathbf{F}} = \mathbf{F}^\top$ and $\bar{\mathcal{L}}_i \mathbf{X} = \mathcal{L}(\mathbf{X}^\top)_i$ for notation simplicity. By using the indicator function, (19) can be reduced to the following optimization problem:

$$\min_{\bar{\mathbf{F}}} \alpha \|\bar{\mathbf{F}}\|_1 + \frac{1}{R} \|\mathbf{M}\mathbf{Z}_{\text{all}}^\top \circ \bar{\mathbf{F}}\|_1 + \iota(\bar{\mathbf{F}}) - \sum_{k=0}^{2^L-1} \log \text{gdet}(\bar{\mathcal{L}}_i(\mathbf{M}^\top \bar{\mathbf{F}})). \quad (22)$$

where ι is defined by

$$\iota(\bar{\mathbf{F}}) = \begin{cases} 0 & \bar{\mathbf{F}} \geq 0 \\ \infty & \text{otherwise.} \end{cases} \quad (23)$$

Owing to the nonnegative constraint on $\bar{\mathbf{F}}$, the first and second terms in (22) can be merged as

$$\min_{\bar{\mathbf{F}}} \frac{1}{R} \|(\alpha \mathbf{H} + \mathbf{M}\mathbf{Z}_{\text{all}}^\top) \circ \bar{\mathbf{F}}\|_1 + \iota(\bar{\mathbf{F}}) - \sum_{k=0}^{2^L-1} \log \text{gdet}(\bar{\mathcal{L}}_i(\mathbf{M}^\top \bar{\mathbf{F}})), \quad (24)$$

where $\mathbf{H} = \mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{E \times 2^{(L+1)-1}}$. By introducing the linear operator $\bar{\mathcal{L}} : \mathbb{R}^{E \times 2^L} \rightarrow \mathbb{R}^{2^L N \times N}$ defined as

$$\bar{\mathcal{L}}(\mathbf{X}) = [\bar{\mathcal{L}}_0(\mathbf{X}), \dots, \bar{\mathcal{L}}_{2^L-1}(\mathbf{X})]^\top, \quad (25)$$

and a dual variable $\mathbf{V} := [\mathbf{V}_0^\top, \dots, \mathbf{V}_{2^L-1}^\top]^\top = \bar{\mathcal{L}}(\mathbf{M}^\top \bar{\mathbf{F}})$, we can convert (24) into the form in (3) as follows:

$$f(\bar{\mathbf{F}}) = 0, \\ g(\bar{\mathbf{F}}) = \frac{1}{R} \|(\alpha \mathbf{H} + \mathbf{M}\mathbf{Z}_{\text{all}}^\top) \circ \bar{\mathbf{F}}\|_1 + \iota(\bar{\mathbf{F}}), \\ h(\mathbf{V}) = - \sum_{k=0}^{2^L-1} \log \text{gdet}(\mathbf{V}_k). \quad (26)$$

The proximal operator for the function g corresponds to that of the weighted ℓ_1 norm with the nonnegative constraint, and it is given by

$$[\text{prox}_{\gamma g}(\mathbf{A})]_{i,j} = \begin{cases} 0 & [\mathbf{A}]_{i,j} \leq \gamma [\mathbf{B}]_{i,j} \\ [\mathbf{A}]_{i,j} - \gamma [\mathbf{B}]_{i,j} & \text{otherwise,} \end{cases} \quad (27)$$

where $\mathbf{B} = \alpha \mathbf{H} + \mathbf{M}\mathbf{Z}_{\text{all}}^\top$.

The proximal operator of h can be computed as follows. In general, the logarithm of a generalized determinant is a nonconvex function. Under the assumption that the learned graph is connected (which is often the case), it can be replaced with a convex function as follows [11, Proposition 1]:

$$\log \text{gdet}(\mathbf{A}) = \log \det(\mathbf{A} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top). \quad (28)$$

Then, the proximal operator is given by

$$\text{prox}_{\gamma(-\log \text{gdet}(\cdot))}(\mathbf{A}) = \mathbf{U} \begin{bmatrix} \phi(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \phi(\lambda_{N-1}) \end{bmatrix} \mathbf{U}^\top, \quad (29)$$

where $\phi(\lambda_i) = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\gamma}}{2}$ and \mathbf{U} and λ_i are the eigenvector matrix and the eigenvalue of $\mathbf{A} + \frac{1}{N} \mathbf{1}\mathbf{1}^\top$, respectively. The eigenvalues are ordered as $\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{N-1}$.¹

Finally, we present the algorithm for the multiresolution TVGL in Algorithm 1. The condition of convergence is given by

$$\gamma_1 \gamma_2 \|\mathbf{M}\tilde{\mathcal{L}}^* \tilde{\mathcal{L}}\mathbf{M}^\top\| < 1. \quad (31)$$

Algorithm 1 Temporal Multiresolution Graph Learning

Input: $\bar{\mathbf{F}}^{(0)}, \mathbf{V}^{(0)}, \{\mathbf{x}_t\}_{t=0}^{T-1}, L, \epsilon$

Output: \mathbf{F}

Divide $\{\mathbf{x}_t\}_{t=0}^T$ into 2^L data segments $\mathbf{X}_0 \dots \mathbf{X}_{2^L-1}$

Compute $\{\mathbf{Z}_k\}_{k=0}^{2^L-1}$ from $\{\mathbf{X}_k\}_{k=0}^{2^L-1}$

while $\|\bar{\mathbf{F}}^{(i+1)} - \bar{\mathbf{F}}^{(i)}\| / \|\bar{\mathbf{F}}^{(i)}\| > \epsilon$ **do**

$\bar{\mathbf{F}}^{(i+1)} := \text{prox}_{\gamma_1 g}(\bar{\mathbf{F}}^{(i)} - \gamma_1 \mathbf{M}\tilde{\mathcal{L}}^* \mathbf{V}^{(i)})$

$\mathbf{V}^{(i)} \leftarrow \mathbf{V}^{(i)} + \gamma_2 \tilde{\mathcal{L}}\mathbf{M}(2\bar{\mathbf{F}}^{(i+1)} - \bar{\mathbf{F}}^{(i)})$

for $k = 0, \dots, 2^L - 1$ **do**

$\mathbf{V}_k^{(i+1)} := \mathbf{V}_k^{(i)} - \gamma_2 \text{prox}_{\frac{1}{\gamma_2}(-\log \text{gdet}(\cdot))} \left(\frac{\mathbf{V}_k^{(i)}}{\gamma_2} \right)$

end for

$\mathbf{V}^{(i+1)} = \begin{bmatrix} \mathbf{V}_0^{(i+1)\top} \\ \vdots \\ \mathbf{V}_{2^L-1}^{(i+1)\top} \end{bmatrix}^\top$

$i \leftarrow i + 1$

end while

Based on the submultiplicativity of the operator norm, the upper-bound of $\|\mathbf{M}\tilde{\mathcal{L}}^* \tilde{\mathcal{L}}\mathbf{M}^\top\|$ can be computed from

$$\|\mathbf{M}\tilde{\mathcal{L}}^* \tilde{\mathcal{L}}\mathbf{M}^\top\| \leq \|\mathbf{M}\| \|\tilde{\mathcal{L}}^* \tilde{\mathcal{L}}\| \|\mathbf{M}^\top\| = \|\tilde{\mathcal{L}}^* \tilde{\mathcal{L}}\| \|\mathbf{M}\mathbf{M}^\top\| = N(2^{L+2} - 2) \quad (32)$$

because of $\|\tilde{\mathcal{L}}^* \tilde{\mathcal{L}}\| = 2N$ and $\|\mathbf{M}\mathbf{M}^\top\| = 2^{L+1} - 1$. Consequently, the convergence condition in (31) can be rewritten as

$$\gamma_1 < \frac{1}{\gamma_2 N(2^{L+2} - 2)}. \quad (33)$$

¹Even if the original graph has disconnected components, we can avoid the problem of the calculation of the proximal operator by adding a small regularizing parameter c to the input as follows [30]:

$$\log \text{gdet}(\mathbf{A}) \approx \log \det(\mathbf{A} + c^2 \mathbf{I}), \quad (30)$$

The proximal operator of this approximation also can be computed in the same manner as (29).

The computational complexity of our algorithm is $O(2^L N^3)$ per iteration.

VI. EXPERIMENTAL RESULTS

In this section, we present experimental results on synthetic and real datasets. The existing and proposed methods are abbreviated as follows:

- SGL based on smoothness criterion (SGL-S) [21].
- SGL with LGMRF (SGL-LG) [11].
- TVGL based on smoothness with temporal variation constraint (TVGL-S) [12], [16], [17].
- TVGL with LGMRF incorporating the temporal variation constraint (TVGL-LG) [18].
- Proposed TMR TVGL (TVGL-MR) described in Section V.

The stopping criterion of the iterations for each methods is set to $\|w^{(n+1)} - w^{(n)}\| / \|w^{(n)}\| < 1.0 \times 10^{-3}$.

A. EXPERIMENTS ON TEMPORAL MULTIREOLUTION GRAPHS

To demonstrate the concept of TVGL for TMR graphs, we first present the results by constructing a simple TMR graph dataset.

1) DATASET

The dataset is constructed in two sequential steps: 1) construction of time-varying graphs and 2) generation of data samples based on the time-varying graphs.

First, we construct TMR graphs with four levels ($l = 0, \dots, 3$) as shown in Fig. 2. The number of vertices N is set to $N = 81$ and the edge weights between vertices are random values drawn from a uniform distribution from the interval $[0.1, 3]$. The lowest resolution graph, i.e., the graph reflecting the global structure, is $\mathbf{W}_{0,0}$, as shown in Fig. 2(a), where the graph has a grid-like structure while the edges only run vertically, except for the horizontal edges at the center of the grid. As shown in Figs. 2(b)–(o), the graphs at levels 1 to 3 have horizontal edges, diagonal edges from the upper right to lower left, and diagonal edges from the upper left to lower right, respectively. By combining $\mathbf{W}_{l,m}$'s, we obtain prototype graphs $\mathbf{W}^{(0)}, \dots, \mathbf{W}^{(7)}$, as shown in Fig. 3.

From the prototype graphs, we then construct time-varying graphs $\{\mathbf{W}_0, \dots, \mathbf{W}_{T-1}\}$. We set $T = 640$ in this experiment. As the number of multiresolution graphs in the highest resolution is eight, each of them has been duplicated 80 times and then they are concatenated, i.e., $\mathbf{W}_t := \mathbf{W}^{(l/80)}$ ($t = 0, \dots, T - 1$).

Second, multivariate time-series signals \mathbf{X} are generated from the following GMRF:

$$\mathbf{x}_t \sim \mathcal{N}(0, (\mathbf{L}_t + \sigma^2 \mathbf{I})^\dagger), \quad (34)$$

where \mathbf{L}_t is the graph Laplacian associated with \mathbf{W}_t . We set σ to 0.5.

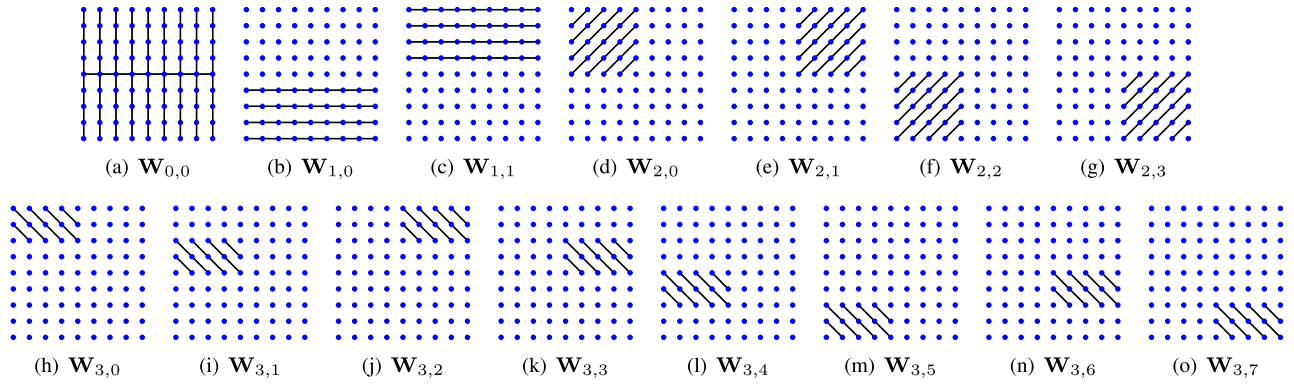


FIGURE 2. Visualization of the ground-truth graphs.

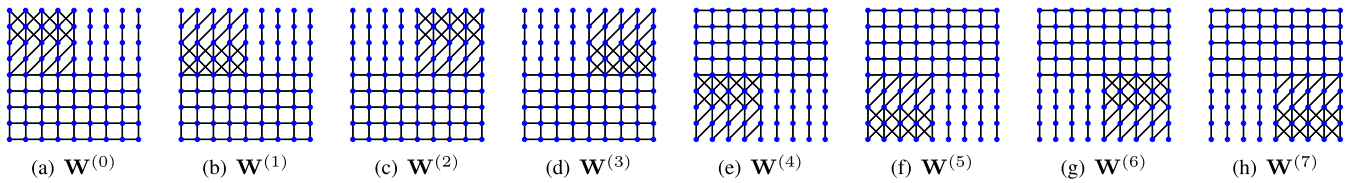


FIGURE 3. Time-varying graphs obtained from the multiresolution graphs in Fig. 2.

2) EXPERIMENTAL CONDITION

We evaluate the performance in terms of relative error and F-measure, each averaged over all time slots. Relative error is given by

$$\text{Relative error} = \frac{\|\widehat{W} - W^*\|_F}{\|W^*\|_F}, \tag{35}$$

where \widehat{W} is the estimated weighted adjacency matrix, and W^* is the ground-truth. It reflects the accuracy of edge weights on the estimated graph.

The F-measure is given by

$$\text{F-measure} = \frac{2\text{tp}}{2\text{tp} + \text{fn} + \text{fp}}, \tag{36}$$

where the true positive (tp) is the number of edges that are included both in \widehat{W} and W^* , the false positive (fn) is the number of edges that are not included in \widehat{W} but are included in W^* , and the false positive (fp) is the number of edges that are included in \widehat{W} but are not included in W^* . The F-measure, which is the harmonic average of the precision and recall, represents the accuracy of the estimated graph topology. The F-measure takes values between 0 and 1. The higher the F-measure, the higher the performance of capturing the graph topology.

In this experiment, we construct training and test data and evaluate the performance of graph learning on the test data using the hyperparameters that minimize the relative error on the training data. We search for optimal hyperparameters using Bayesian optimization [39]. Additionally, ℓ_1 norm is used for the temporal variation regularization of the existing TVGL approaches.

We evaluate the performance with different window sizes to study the robustness of each method for the choice of the window size K . The existing methods use $K = 20, 40,$ or $80,$ and the proposed method uses the maximum temporal resolution level $L = 5$. The proposed method can reconstruct time-varying graphs corresponding to $K = \{20, 40, 80\}$ from a set of TMR graphs. Note that the existing methods need to fix K before running their algorithms, whereas the proposed TVGL method simultaneously estimates time-varying graphs in the different window sizes.

3) RESULTS

Table 1 summarizes the average performance of the learned graphs. As shown in the table, TVGL-MR nearly outperforms the other methods both in terms of F-measure and relative error. This indicates that the TVGL performances can be improved by TVGL-MR if time-varying graphs can be assumed to have multiresolution characteristics.

Fig. 4 visualizes the time-varying graphs learned by TVGL-S, TVGL-LG, and TVGL-MR. As shown in Fig. 4, the alternative TVGL methods fail to capture temporal multiresolution structures, particularly those at the high-resolution level. In contrast, the proposed method captures edges localized at various temporal resolutions.

Furthermore, Fig. 5 shows the TMR graphs learned by TVGL-MR. The figure also demonstrates that the proposed method can successfully learn TMR graphs.

B. EXPERIMENTS ON SINGLE RESOLUTION GRAPHS

The previous experiment demonstrates the effectiveness of the proposed method for TMR graphs. While the proposed

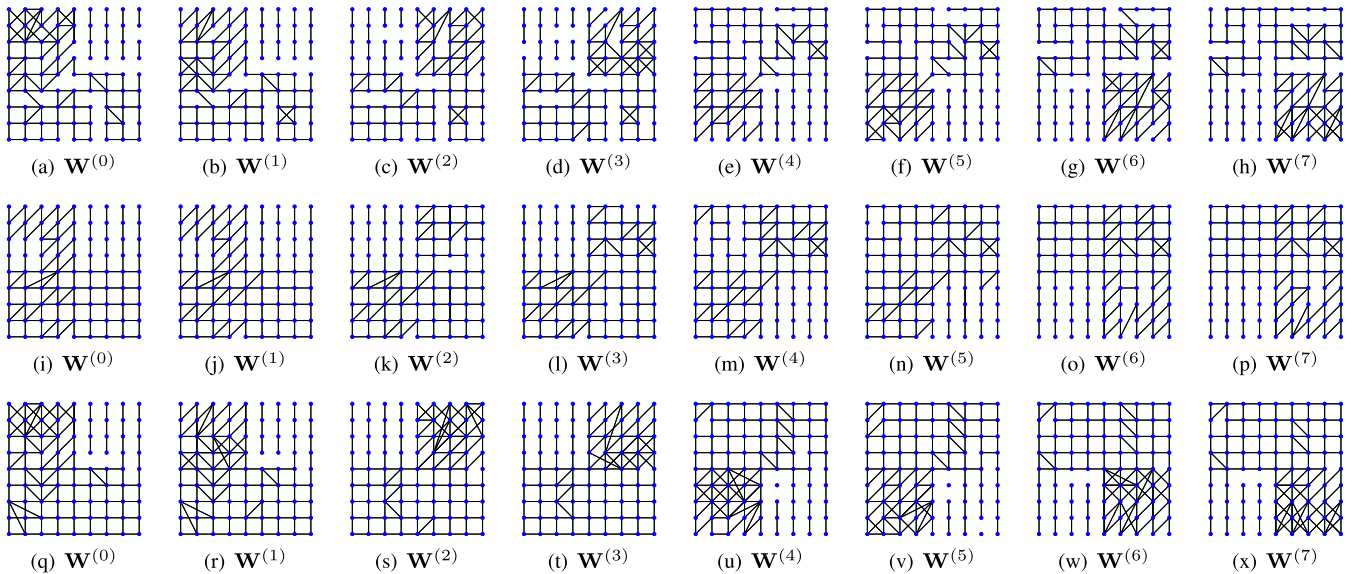


FIGURE 4. Visualization of time-varying graphs. The top, middle, and bottom rows represent the time-varying graphs learned by TVGL-S, TVGL-LG, and TVGL-MR, respectively.

TABLE 1. Comparison of the performance for learning time-varying graph.

Methods	Window size	F-measure	Relative Error
SGL-S	20	0.571	0.710
	40	0.717	0.624
	80	0.790	0.581
	avg.	0.693	0.638
SGL-LG	20	0.596	0.730
	40	0.726	0.576
	80	0.831	0.431
	avg.	0.718	0.579
TVGL-S	20	0.798	0.479
	40	0.817	0.508
	80	0.855	0.431
	avg.	0.823	0.472
TVGL-LG	20	0.890	0.370
	40	0.888	0.419
	80	0.871	0.513
	avg.	0.883	0.434
TVGL-MR	20	0.842	0.415
	40	0.895	0.341
	80	0.915	0.314
	avg.	0.884	0.357

method is not specifically designed for learning single resolution TVGL, here, we compare TVGL performances with the other methods for some single resolution time-varying graphs.

1) DATASETS

The dataset is constructed with the same steps described in Section VI-A. In this experiment, we construct two types of time-varying graphs as follows:

a: EDGE-MARKOVIAN EVOLVING GRAPH (EMEG)

EMEG is a stochastic time dependency evolving graph [40]. Each edge in EMEG follows the Markovian process. EMEG

$\mathcal{G}_s = \{\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{W}_t)\}$ is satisfied as

$$\begin{cases} p(e \in \mathcal{E}_{t+1} | e \notin \mathcal{E}_t) = q_1 \\ p(e \notin \mathcal{E}_{t+1} | e \in \mathcal{E}_t) = q_2 \end{cases} \quad (37)$$

where q_1 and q_2 are called *birth rate* and *death rate*, respectively. We generate an Erdős–Rényi graph with $N = 36$, $p = 0.1$ as the initial graph \mathcal{G}_0 . The edge weights of the initial graph are selected from the uniform distribution with the interval $[0.1, 3]$, and the weights of the newborn edges are also selected from the same distribution. We set $q_1 = 0.001$ and $q_2 = 0.01$.

b: SWITCHING BEHAVIOR GRAPH (SBG)

SBG is a time-varying graph that exhibits the transition of connectivity states. It often appears in brain connectivity dynamics [41], [42]. We construct an SBG using the following procedure. We generate six static graphs used as the connectivity states. Each of the graphs is initialized to an Erdős–Rényi graph with $N = 36$, an edge connection probability $p = 0.05$, and edge weights drawn from a uniform distribution in the interval $[0.1, 3]$. The initial state is selected randomly from the six connectivity states, and its state remains with a 98% probability and transits to another connectivity state with the 2% probability at each time.

c: GENERATING GRAPH SIGNALS

Given graph Laplacians $\mathbf{L}^{(0)}, \dots, \mathbf{L}^{(127)}$ of the constructed time-varying graphs, we generate multivariate time-series signal $\mathbf{x}_0, \dots, \mathbf{x}_{5119}$ from the following GMRF:

$$\mathbf{x}_t \sim \mathcal{N}(0, (\mathbf{L}^{(t/40)} + \sigma^2 \mathbf{I})^\dagger), \quad (38)$$

where σ^2 is the variance of the white Gaussian noise. We set $\sigma = 0.5$ in this experiment.

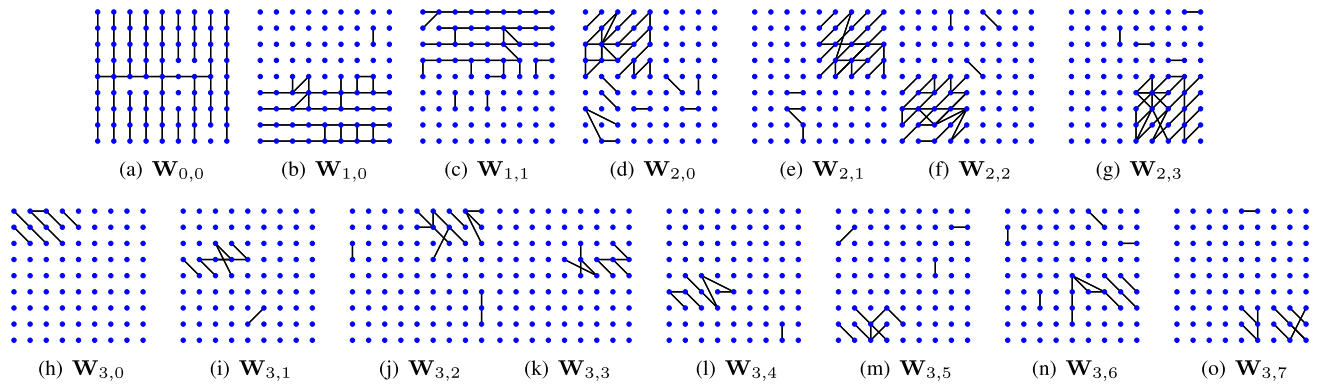


FIGURE 5. Visualization of the TMR graphs learned by TVGL-MR.

d: REGULARIZATION FUNCTIONS FOR ALTERNATIVE TVGL METHODS

TVGL methods, i.e., TVGL-S and TVGL-LG, require choosing the regularization function based on the prior knowledge of temporal graph evolution. For EMEG and SBG, we adopt ℓ_1 and $\ell_{2,1}$ -norm as the possible regularization functions, respectively.

2) RESULTS

Table 2 summarizes the performances of SGL/TVGL methods on different datasets. TVGL methods outperform the static methods on all datasets. This implies that the regularization for the temporal graph evolution or TMR assumption improves the graph learning performance.

Among the TVGL methods, TVGL-MR ranks first or second in this experiment. This suggests the effectiveness and robustness of the proposed method even for single resolution time-varying graphs. It is also worth noting that, TVGL-MR can exhibit performance comparable to that of time-varying methods without the prior knowledge of the graph evolution over time, i.e., the regularization function. Typically, existing TVGL approaches require *both* prior knowledge and hyperparameter(s). In contrast, the only assumption in the proposed method is that time-varying graphs are characterized by the multiresolution property, which is a natural assumption of signal processing. This implies the flexibility of the proposed method.

Figs. 6 and 7 show the visualization of the temporal variation in the ground-truth graphs and the learned graphs with a window size of 40, respectively. The vertical and horizontal axes of these figures represent the edge and time slot indices of the time-varying graph, and the color represents the intensity of the edge weights. For simple visualization, the first 100 edge indices are visualized.

As can be seen in Fig. 7, SGL-S and SGL-LG lose the temporal relations, whereas TVGL-S, TVGL-LG, and TVGL-MR can capture the original structures more precisely than static methods. Time-varying graphs by TVGL-S, TVGL-LG, and TVGL-MR are similar, but the proposed method tends to yield larger edge weights.

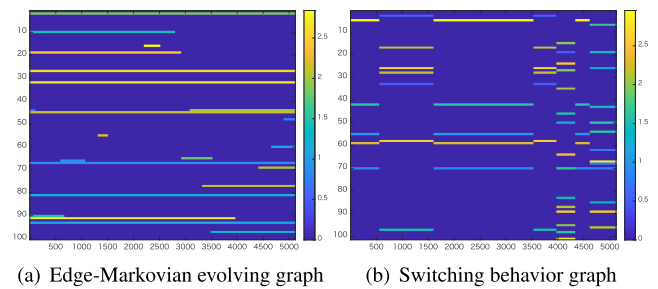


FIGURE 6. Visualization of the temporal variations in the ground-truth time-varying graph of each dataset.

C. LEARNING TEMPORAL MULTIREOLUTION GRAPHS FROM REAL TEMPERATURE DATA

Finally, we apply TVGL-MR to the real temperature data in Hokkaido, the northernmost island in Japan. The goal of this experiment is to explore the common (time-invariant) and seasonal relationships among geographical regions using the proposed method.

We use the average temperature data² measured at 172 recording locations in Hokkaido from March 2014 to February 2015. We perform TVGL-MR with $L = 3$ (i.e., the number of graphs is four at the highest level).

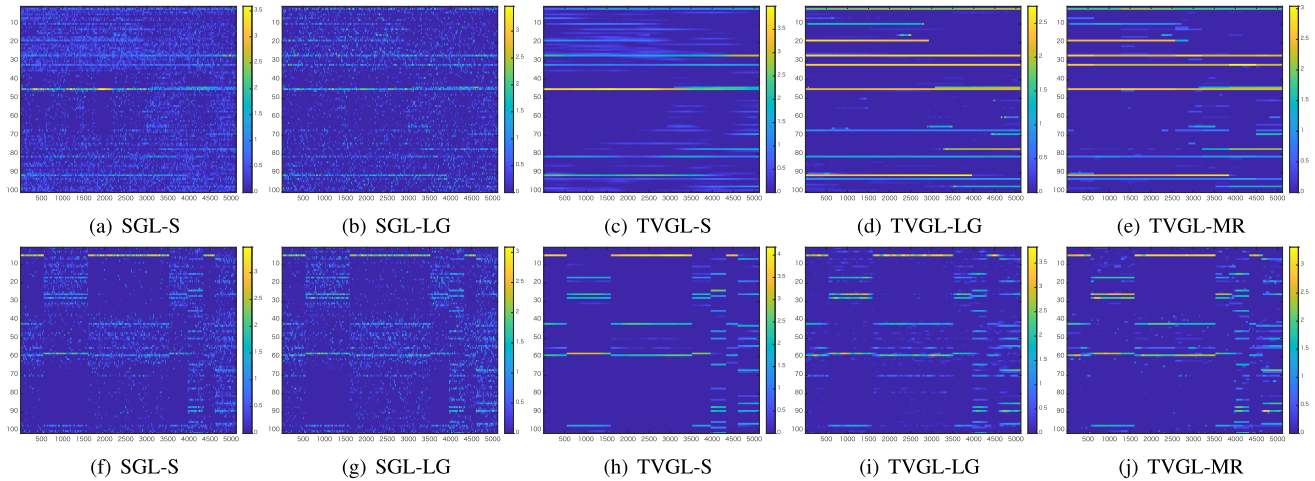
Fig. 8 shows the lowest resolution graph $W_{0,0}$ obtained by TVGL-MR and the graph obtained by SGL-LG from data of all time slots. Note that both of them can be regarded as static graphs. Focusing on the graph learned by TVGL-MR, the following characteristics are observed:

- Vertices close to each other are basically connected, and edges between closer nodes tend to have large weights. However, if the recording locations are separated by a mountain (brownish area), nodes may not be connected even if they are geographically close.
- Vertices with similar geographic features are often connected, i.e., ones along the coast are connected to each

²The Japan Meteorological Agency provided the daily temperature data from their website at <https://www.jma.go.jp/jma/index.html>

TABLE 2. Comparison of the F-measure and relative error for learning time-varying graph. The bold and underlined values represent the best and second-best performance among the methods, respectively.

F-measure						
Dataset	WS	SGL-S	SGL-LG	TVGL-S	TVGL-LG	TVGL-MR
EMEG	10	0.320	0.367	0.525	<u>0.710</u>	0.800
	20	0.389	0.448	0.523	<u>0.748</u>	0.776
	40	0.495	0.530	0.518	<u>0.727</u>	0.772
	avg.	0.401	0.448	0.522	<u>0.601</u>	0.783
SBG	10	0.303	0.313	0.804	0.418	<u>0.754</u>
	20	0.449	0.382	0.800	0.579	<u>0.722</u>
	40	0.524	0.436	0.818	0.664	<u>0.696</u>
	avg.	0.425	0.377	0.807	0.557	<u>0.724</u>
Relative error						
EMEG	10	0.885	0.906	0.537	0.191	<u>0.296</u>
	20	0.874	0.832	0.532	0.196	<u>0.262</u>
	40	0.720	0.713	0.527	<u>0.269</u>	0.254
	avg.	0.826	0.817	0.532	0.219	<u>0.271</u>
SBG	10	0.882	0.888	<u>0.419</u>	0.571	0.410
	20	0.777	0.774	<u>0.399</u>	0.349	<u>0.376</u>
	40	0.658	0.618	0.402	0.354	<u>0.359</u>
	avg.	0.772	0.760	<u>0.406</u>	0.424	0.382

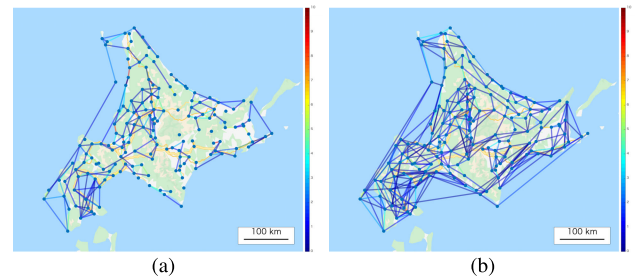
**FIGURE 7.** Visualization of the temporal variations in the learned time-varying graph. The top row and bottom rows depict the variations for EMEG and SBG datasets, respectively. The colors in these figures represent the weights of the edges.

other, and the similar characteristic is observed for inland vertices.

The above-mentioned characteristics seem reasonable because the relationship based on the distance between nodes or geographic features is static.

In contrast, the graph learned by SGL-LG is denser than that by TVGL-MR and includes many edges connecting distant nodes. Such edges may be derived from the seasonal behavior, which is described later. As SGL-LG learns a static graph from all the time slots without separating structures localized at various temporal resolutions, the learned graph may include both common and seasonal edges.

Fig. 9 shows $\mathbf{W}_{2,0}, \dots, \mathbf{W}_{2,3}$ learned by TVGL-MR, which corresponds to season-specific graphs. In contrast to the static graph, these seasonal graphs have few edges connecting nodes close to each other. This suggests that the distance-based relationship would have a weak effect

**FIGURE 8.** Visualization of learned graphs. (a) $\mathbf{W}_{0,0}$ learned by the TVGL-MR. (b) Graph learned by SGL-LG from data of all time slots.

on the seasonal behavior. Furthermore, the summer- and winter-specific graphs have more edges than those of the spring and autumn-specific graphs. This seems intuitive because the seasonal effects in summer and winter are

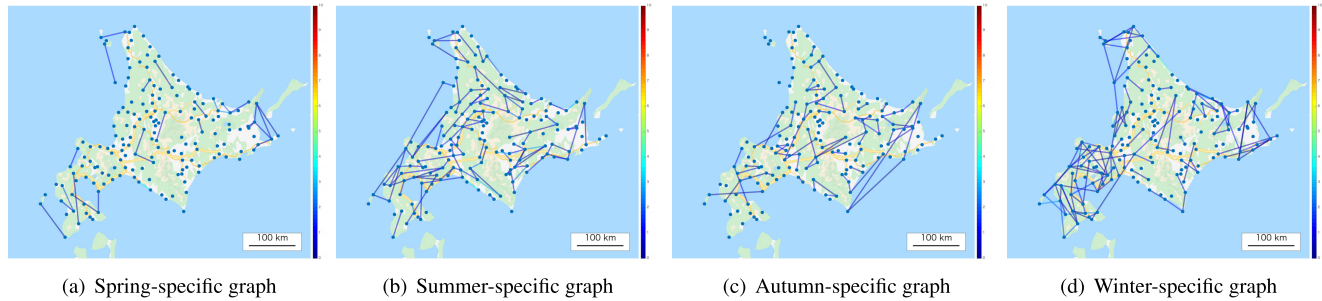


FIGURE 9. Visualization of the season-specific graphs learned by TVGL-MR: (a)–(d) corresponds to $W_{2,0}, \dots, W_{2,3}$, respectively.

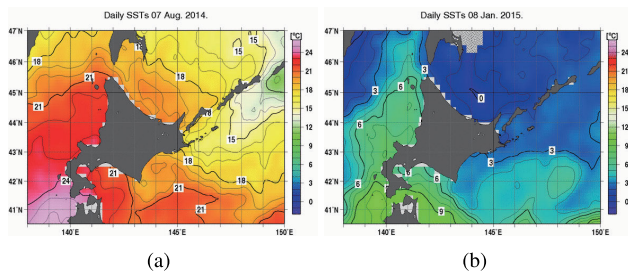


FIGURE 10. Daily sea surface temperature. (a) August 7, 2014. (b) January 8, 2015.

expected to be stronger than those in mild, such as spring and fall.

Furthermore, edges connecting distant coastal nodes in the summer and winter-specific graphs (which are also observed in SGL-LG in Fig. 8(b)) can be attributed to the effects of seasonal sea currents. Fig. 10 shows the sea surface temperature (SST)³ on August 7, 2014, and January 8, 2015. As can be seen in Figs. 9(b), 9(d), and 10, vertices connected along coasts in the summer- and winter-specific graphs reflect SST behaviors for the two seasons.

VII. CONCLUSION

We proposed a temporal multiresolution graph learning method from multivariate time-series data. The proposed method is designed based on a signal generation model in accordance with an LGMRF, and enables the capture of time-varying structures having a multiresolution property in one single framework. The TVGL is formulated as a convex optimization problem and can be solved efficiently using a primal-dual splitting algorithm. The experiments on synthetic and real datasets demonstrate that the proposed method outperforms the existing static and time-varying graph learning methods.

REFERENCES

- [1] 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.
- [2] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.
- [3] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [4] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling signals on graphs: From theory to applications," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 14–30, Nov. 2020.
- [5] W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Graph frequency analysis of brain signals," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1189–1203, Oct. 2016.
- [6] H. Higashi, T. M. Rutkowski, T. Tanaka, and Y. Tanaka, "Multilinear discriminant analysis with subspace constraints for single-trial classification of event-related potentials," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1295–1305, Oct. 2016.
- [7] S. Ono, I. Yamada, and I. Kumazawa, "Total generalized variation for graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2015, pp. 5456–5460.
- [8] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137–148, Jun. 2016.
- [9] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, "Graph spectral image processing," *Proc. IEEE*, vol. 106, no. 5, pp. 907–930, May 2018.
- [10] I. Jabłoński, "Graph signal processing in applications to sensor networks, smart grids, and smart cities," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7659–7666, Dec. 2017.
- [11] 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.
- [12] 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.*, May 2019, pp. 5411–5415.
- [13] 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.
- [14] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. IEEE*, vol. 106, no. 5, pp. 787–807, May 2018.
- [15] 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.
- [16] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 2826–2830.
- [17] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning with constraints on graph temporal variation," 2020, *arXiv:2001.03346*. [Online]. Available: <http://arxiv.org/abs/2001.03346>
- [18] J. V. de Miranda Cardoso and D. P. Palomar, "Learning undirected graphs in financial markets," 2020, *arXiv:2005.09958*. [Online]. Available: <http://arxiv.org/abs/2005.09958>
- [19] K. Yamada and Y. Tanaka, "Learning graphs with multiple temporal resolutions," in *Proc. APSIPA ASC*, 2020, pp. 139–142.
- [20] 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.

³The daily SST was provided by Japan Meteorological Agency, from their website at <https://www.jma.go.jp/jma/index.html>

- [21] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 920–929.
- [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.*, Mar. 2017, pp. 6508–6512.
- [23] 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, Jun. 2019.
- [24] S. Kumar, J. Ying, J. V. de Miranda Cardoso, and D. Palomar, "Structured graph learning via Laplacian spectral constraints," in *Advances in Neural Information Processing Systems*, vol. 32. Red Hook, NY, USA: Curran Associates, 2019, pp. 11647–11658.
- [25] 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.
- [26] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.
- [27] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 484–499, Sep. 2017.
- [28] S. Segarra, Y. Wang, C. Uhler, and A. G. Marques, "Joint inference of networks from stationary graph signals," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 975–979.
- [29] M. Navarro, Y. Wang, A. G. Marques, C. Uhler, and S. Segarra, "Joint inference of multiple graphs from matrix polynomials," 2020, *arXiv:2010.08120*. [Online]. Available: <http://arxiv.org/abs/2010.08120>
- [30] H. P. Maretic and P. Frossard, "Graph Laplacian mixture model," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 261–270, 2020.
- [31] J.-Y. Kao, D. Tian, H. Mansour, A. Ortega, and A. Vetro, "Disc-GLasso: Discriminative graph learning with sparsity regularization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 2956–2960.
- [32] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, "Network inference via the time-varying graphical lasso," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 205–213.
- [33] S. Yang, Z. Lu, X. Shen, P. Wonka, and J. Ye, "Fused multiple graphical lasso," *SIAM J. Optim.*, vol. 25, no. 2, pp. 916–943, 2015.
- [34] L. Gan, X. Yang, N. Narisetty, and F. Liang, "Bayesian joint estimation of multiple graphical models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 9802–9812.
- [35] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *J. Roy. Statist. Soc. B, Statist. Methodol.*, vol. 76, no. 2, pp. 373–397, Mar. 2014.
- [36] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [37] L. Condat, "A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, Aug. 2013.
- [38] A. Holbrook, "Differentiating the pseudo determinant," *Linear Algebra Appl.*, vol. 548, pp. 293–304, Jul. 2018.
- [39] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," in *Proc. Uncertainty Artif. Intell.*, 2014, pp. 250–259.
- [40] A. E. F. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri, "Flooding time in edge-Markovian dynamic graphs," in *Proc. 27th ACM Symp. Princ. Distrib. Comput.* New York, NY, USA: Association for Computing Machinery, 2008, pp. 213–222.
- [41] E. A. Allen, E. Damaraju, S. M. Plis, E. B. Erhardt, T. Eichele, and V. D. Calhoun, "Tracking whole-brain connectivity dynamics in the resting state," *Cerebral Cortex*, vol. 24, no. 3, pp. 663–676, 2014.
- [42] J. Taghia, S. Ryali, T. Chen, K. Supekar, W. Cai, and V. Menon, "Bayesian switching factor analysis for estimating time-varying functional connectivity in fMRI," *NeuroImage*, vol. 155, pp. 271–290, Jul. 2017.



KOKI YAMADA (Member, IEEE) received the B.S. degree in physics and informatics and the M.S. degree in science and engineering from Yamaguchi University, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Graduate School of EECS, Tokyo University of Agriculture and Technology. His current research interests include graph signal processing and high-dimensional data analysis.



YUICHI TANAKA (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 2003, 2005, and 2007, respectively. From 2006 to 2008, he was also a Visiting Scholar with the University of California, San Diego. He was a Postdoctoral Scholar at Keio University, from 2007 to 2008, and supported by the Japan Society for the Promotion of Science (JSPS). From 2008 to 2012, he was an Assistant

Professor with the Department of Information Science, Utsunomiya University, Tochigi, Japan. Since 2012, he has been an Associate Professor with the Department of Electrical Engineering and Computer Science, Tokyo University of Agriculture and Technology, Tokyo, Japan. He currently has a cross appointment as a PRESTO Researcher with the Japan Science and Technology Agency. His current research interests include the field of high-dimensional signal processing and machine learning which includes graph signal processing, geometric deep learning, sensor networks, image/video processing in extreme situations, biomedical signal processing, and remote sensing. He is an Elected Member of the APSIPA Signal and Information Processing Theory and Methods (SIPTM) and Image, Video and Multimedia (IVM) Technical Committees. He was a recipient of the Yasujiro Niwa Outstanding Paper Award, in 2010, the TELECOM System Technology Award, in 2011, and the Ando Incentive Prize for the Study of Electronics, in 2015. He also received the IEEE Signal Processing Society Japan Best Paper Award, in 2016, and Best Paper Awards in APSIPA ASC 2014 and 2015. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, from 2016 to 2020, and the *IEICE Transactions on Fundamentals*, from 2013 to 2017.

• • •