Time-Varying Graph Learning for Data with Heavy-Tailed Distribution

Amirhossein Javaheri *Graduate Student Member IEEE*, Jiaxi Ying *Member IEEE*, Daniel P. Palomar, *Fellow, IEEE*, and Farokh Marvasti, *Life Senior Member, IEEE*

***Abstract*—Graph models provide efficient tools to capture the underlying structure of data defined over networks. Many real-world network topologies are subject to change over time. Learning to model the dynamic interactions between entities in such networks is known as time-varying graph learning. Current methodology for learning such models often lacks robustness to outliers in the data and fails to handle heavy-tailed distributions, a common feature in many real-world datasets (e.g., financial data). This paper addresses the problem of learning time-varying graph models capable of efficiently representing heavy-tailed data. Unlike traditional approaches, we incorporate graph structures with specific spectral properties to enhance data clustering in our model. Our proposed method, which can also deal with noise and missing values in the data, is based on a stochastic approach, where a non-negative vector auto-regressive (VAR) model captures the variations in the graph and a Student-t distribution models the signal originating from this underlying time-varying graph. We propose an iterative method to learn time-varying graph topologies within a semi-online framework where only a mini-batch of data is used to update the graph. Simulations with both synthetic and real datasets demonstrate the efficacy of our model in analyzing heavy-tailed data, particularly those found in financial markets.**

***Index Terms*—Time-varying, graph learning, Laplacian matrix, data clustering, heavy-tailed distribution, corrupted measurements, financial data**

I. Introduction

GRAPH signal processing (GSP) is an interesting research area that combines graph theory and signal processing to model and analyze signals defined on graph structures, which has numerous real-world applications in social networks [2], image processing [3], data mining, communications [4], finance [5], and more. The process of inferring the structure of a graph from data is known as graph learning [6]. There are generally two types of graphs, namely undirected and directed, with each type modeling different characteristics of data. An undirected graph models bilateral relationships or similarities in data, while a directed graph is commonly used to model unilateral causal dependencies [7].

There are various approaches in the literature for modeling signals via graphs. Many graph learning methods consider a probabilistic model for the data, in which a graph structure represents the statistics of the data. A foundational technique in this domain is the Gaussian Markov Random Field (GMRF) model [8], which assumes that the data follows a multivariate Gaussian distribution. In a general GMRF, the precision matrix encodes the conditional independence relationships between variables, thereby defining the structure of the underlying undirected graph. Recently, there has been growing interest in Laplacian constrained GMRFs [9]–[11], where the precision matrix is specifically constrained to be the Laplacian matrix of an undirected graph [12]. This constraint is particularly desirable for modeling smooth signals on graphs, where a higher weight between two nodes signifies a stronger similarity between their signal values [13]. Under this framework, the graph (the Laplacian matrix) can be inferred via maximum likelihood (ML) or maximum a posteriori (MAP) estimation. The Graphical LASSO (GLASSO) [14] is one of the early works in this regard, which was later improved by introducing structural [11], [15]–[18] and spectral constraints [19] into the problem of learning the graph from data. Directed graph topologies, used for modeling directional dependencies, can also be learned using different approaches. Some topology identification methods are based on structural equation models [20], while many other approaches use vector auto-regressive (VAR) models that can be represented with directed graphs [21]–[23]. There have also been some recent works that incorporate both directed and undirected graph topologies to model spatial and temporal correlations at the same time [24].

The methodologies mentioned above are mostly applicable for learning static graphs. However, in many real-world applications, such as social networks or finance, the network structure is subject to change over time. Therefore, one of the challenges in graph learning is to learn a time-varying graph topology. There are several approaches in the literature for learning such dynamic graphs [25]–[30]. However, the existing approaches lack robustness to data outliers and cannot efficiently model heavy-tailed distributions, such as those observed in financial data. Moreover, these models also fail to capture graph topologies that adhere to specific structural and spectral properties, such as k-component graphs. In this paper, we propose a novel framework for learning time-varying graphs that incorporates spectral and structural constraints

This work was supported by the Hong Kong GRF 16206123 research grant and the Hong Kong RGC Postdoctoral Fellowship Scheme of Project No. PDFS2425-6S05. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yanning Shen. (Corresponding author: Amirhossein Javaheri.)

Amirhossein Javaheri is with the Hong Kong University of Science and Technology and Sharif University of Technology (e-mail: sajavaheri@connect.ust.hk, javaheri_amirhossein@ee.sharif.edu). Jiaxi Ying and Daniel P. Palomar are with the Hong Kong University of Science and Technology, Hong Kong (e-mails: jx.ying@connect.ust.hk, palomar@ust.hk), Farokh Marvasti is with Sharif University of Technology, Iran (e-mail: marvasti@sharif.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material includes supporting lemmas and proofs. This material is 234 KB in size.. A conference version of this paper is presented at the 2024 EUSIPCO conference [1].

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173

in the problem of inferring graph topologies. Our approach is based on a stochastic model that can first characterize the statistical properties of heavy-tailed data and second be employed for data clustering. We subsequently investigate the applications of our framework for time-varying data analysis in financial markets.

II. Overview of Existing Works

A. Notation

In this paper, bold lower-case letters are employed for vectors (e.g., $\mathbf{x}$) and bold upper-case letters are used to denote matrices (e.g., $\mathbf{X}$). The operator $\det^\star$ represents the generalized determinant (the product of non-zero eigenvalues of a matrix), and $\mathbb{1}(\cdot)$ denotes the identity operator. The Hadamard (point-wise) product and division are respectively denoted by $\odot$ and $\oslash$. We also use $\|\mathbf{x}\|_p$ for the ℓ_p vector norm (simply omitting the subscript for $p=2$) and $\|\mathbf{X}\|_F$ for the matrix Frobenius norm. The notations $\|\mathbf{X}\|_{1,\text{off}}$ and $\|\mathbf{X}\|_{F,\text{off}}$ respectively denote the ℓ_1 norm and the Frobenius norm of the off-diagonal elements of the matrix $\mathbf{X}$. The $\ell_{p,q}$ norm of a matrix is denoted with $\|\mathbf{X}\|_{p,q}$. We use $\text{diag}(\mathbf{X})$ to denote the vector of diagonal elements of $\mathbf{X}$ and $\text{Diag}(\mathbf{x})$ to denote a diagonal matrix with $\mathbf{x}$ on its diagonals. The graph Laplacian, the adjacency, and the degree operators [19] are respectively denoted by $\mathcal{A}$, $\mathcal{L}$, and $\mathfrak{d}$. We denote the ground-truth number of clusters in the data with uppercase letter K and the rank constraint parameter of our method with lowercase letter k.

B. Problem Statement

Assume a dynamic signal $\mathbf{x}_t \in \mathbb{R}^p$ defined over a time-varying undirected graph structure, where each vertex of the graph represents an element of the signal, and the edge weights encode the (bilateral) interactions between these elements. Suppose the weights of the graph vary only at specific time instants (piece-wise constant graph). We may then segment the time indices into frames of length T_n, where the graph weights are assumed to remain constant within each time frame. Let F_n denote the time indices of the n-th frame of data, where $n \in \{1,\dots,N\}$. In the case of no overlap between consecutive frames, as shown in Fig. 2, we have $T = \sum_{n=1}^{N} T_n$. A time-varying undirected graph structure can subsequently be

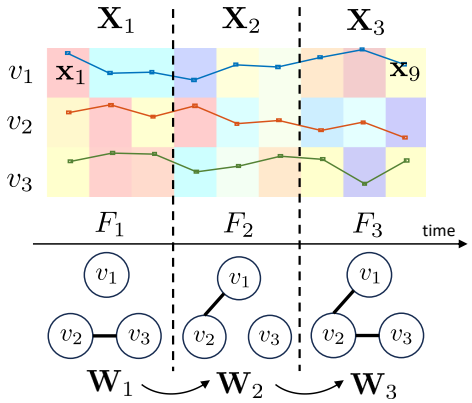


Fig. 1: Illustration of the concept of time-varying graphs.

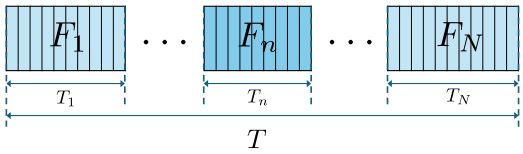


Fig. 2: Illustration of the time frames.

represented by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}_n, \mathbf{W}_n\}$. Here, $\mathcal{V} = \{1,\dots,p\}$ denotes the set of vertices (which remains fixed over time), $\mathcal{E}_n \subseteq \{\{i,j\} \mid i,j \in \mathcal{V}\}$ represents the time-varying set of edges (unordered pairs of nodes connected to each other) at time frame n, and $\mathbf{W}_n$ is the (weighted) adjacency matrix. Each entry $W_{i,j}(n) \geq 0$ of $\mathbf{W}_n$ denotes the weight of the edge between vertex i and vertex j during time frame n. Alternatively, one can represent an undirected graph by the vector of all possible edge weights $\mathbf{w}_n \in \mathbb{R}_+^{p(p-1)/2}$. The edge weights can also be mapped to the adjacency matrix using the adjacency operator $\mathcal{A}$ [19], i.e., $\mathbf{W}_n = \mathcal{A}\mathbf{w}_n$. Furthermore, the time-varying Laplacian matrix can be derived as $\mathbf{L}_n = \text{Diag}(\mathbf{W}_n\mathbf{1}) - \mathbf{W}_n = \mathcal{L}\mathbf{w}_n$, where $\mathcal{L}$ denotes the Laplacian operator [19].

Consider T snapshots (time measurements) of the signal vertically arranged in the columns of $\mathbf{X} = [\mathbf{x}_1,\dots,\mathbf{x}_T] \in \mathbb{R}^{p\times T}$. Let $\mathbf{X}_n = [\mathbf{x}_t \mid t \in F_n]$ denote the data matrix at time frame n.

A time-varying graph learning problem can be generally formulated as follows:

$$\min_{\{\mathbf{S}_n|_{n\in\mathcal{T}}\}\in\Omega_{\mathbf{S}}} \sum_{n\in\mathcal{T}} f_1(\mathbf{S}_n,\boldsymbol{\Sigma}_n) + f_2(\mathbf{S}_n) + f_3(\mathbf{S}_n,\mathbf{S}_{n-1}), \tag{1}$$

where $\mathbf{S}_n$ denotes the graph matrix (Laplacian or the weighted adjacency matrix), $\Omega_{\mathbf{S}}$ denotes the set of feasible graph matrices, $\mathcal{T} \subseteq \{1,\cdots,N\}$ denotes the set of frame indices for which the data is available, and $\boldsymbol{\Sigma}_n$ denotes the data statistics matrix (e.g., sample covariance matrix). Here $f_1(\cdot)$ is a fidelity criterion measuring how well the graph matches the statistics of the data, $f_2(\cdot)$ is a regularization function used to promote properties such as sparsity, and $f_3(\cdot)$ is a temporal consistency term formulating the smoothness of the graph variations.

In the next part, we will discuss different choices for the objective function proposed by current methods.

C. Related Works

The notion of time-dependent graphs has roots dating back to the late 1990s [31]; however, the *time-varying graph learning* methodology and concept were quite recently established. The time-varying graphical LASSO (TVGLASSO) [26] is one of the early works on this topic, which extends the well-known GLASSO [14] inference method to the case of time-varying topologies. Let $\mathbf{L}_n$ denote the Laplacian of a time-varying graph at frame n. Then, the graph learning problem in this paper is formulated as follows:

$$\min_{\{\mathbf{L}_n|_{n=1}^N\}\succ\mathbf{0}} \sum_{n=1}^{N} \Big(-T_n \log\det(\mathbf{L}_n) + T_n \,\text{tr}(\boldsymbol{\Sigma}_n\mathbf{L}_n) + \lambda \|\mathbf{L}_n\|_{1,\text{off}} \Big) + \beta \sum_{n=2}^{N} h(\mathbf{L}_n - \mathbf{L}_{n-1}), \tag{2}$$

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173

where $\mathbf{\Sigma}_n$ denotes the data statistics matrix at time frame n (usually the sample covariance matrix) used as a similarity criterion. Moreover, $h(\cdot)$ is a regularization function utilized as a measure of the temporal smoothness of the graph variations. The TVGLASSO, however, does not incorporate the structural constraints of the combinatorial graph Laplacian (CGL) matrix [11], and the learned matrices $\{\mathbf{L}_n \,|_{n=1}^N\}$ are only positive definite inverse covariance (precision) matrices. Another early work on time-varying graph topology identification is the method in [25]. This work incorporates the Laplacian structural constraints by reformulating the problem in terms of the weighted adjacency matrices $\{\mathbf{W}_n \,|_{n=1}^N\}$ as follows:

$$\begin{aligned}\min_{\{\mathbf{W}_n|_{n=1}^N\}\geq\mathbf{0}} \sum_{n=1}^N \left(\mathrm{tr}(\mathbf{W}_n\mathbf{Z}_n^\top) + f_W(\mathbf{W}_n)\right) \\ +\gamma\sum_{n=2}^N h(\mathbf{W}_n,\mathbf{W}_{n-1}),\end{aligned} \tag{3}$$

where $\mathbf{Z}_n$ denotes the distance matrix whose (i,j)-th entry equals the distance between signal elements i and j at time frame n. For Euclidean distance, $[\mathbf{Z}_n]_{i,j} = \left\|\tilde{\mathbf{x}}_{n_i} - \tilde{\mathbf{x}}_{n_j}\right\|$, where $\tilde{\mathbf{x}}_{n_i}$ denotes the i-th row of the data matrix $\mathbf{X}_n$. Here, $f_W(\cdot)$ is a measure of graph smoothness, and $h(\cdot)$ is a regularization function for the temporal variations of the graph weights. Specifically, $f_W(\mathbf{W}) = -\alpha\mathbf{1}^\top\log(\mathbf{W}\mathbf{1}) + \beta\left\|\mathbf{W}\right\|_F^2$ and $h(\mathbf{W}_n,\mathbf{W}_{n-1}) = \left\|\mathbf{W}_n - \mathbf{W}_{n-1}\right\|_F^2$ are used in this paper. Another variant of the above formulation with $h(\mathbf{W}_n,\mathbf{W}_{n-1}) = \left\|\mathbf{W}_n - \mathbf{W}_{n-1}\right\|_{1,1}$ is also considered in [32], which is solved using a primal-dual splitting method.

In [27], the static factor graph model in [33] is generalized to the time-varying graph factor analysis (TGFA) framework, where the graph signal at time-stamp t is considered to have a Gaussian distribution as follows:

$$\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{L}_n^\dagger + \sigma_\epsilon^2\mathbf{I}), \quad \forall t \in F_n. \tag{4}$$

The problem is then formulated as

$$\begin{aligned}\min_{\{\mathbf{L}_n|_{n=1}^N\}\in\Omega_\mathbf{L}} \sum_{n=1}^N \left(\mathrm{tr}(\mathbf{L}_n\mathbf{\Sigma}_n^\top) + f_L(\mathbf{L}_n)\right) \\ +\eta\sum_{n=2}^N R(\Delta\mathbf{L}_n\odot\mathbf{H}),\end{aligned} \tag{5}$$

where $\mathbf{H} = \mathbf{I} - \mathbf{1}\mathbf{1}^\top$, $\Delta\mathbf{L}_n = \mathbf{L}_n - \mathbf{L}_{n-1}$, and $\Omega_\mathbf{L}$ denotes the set of feasible CGL Laplacian matrices, with $f_L(\mathbf{L}_n) = -\alpha\mathbf{1}^\top\log(\mathrm{diag}(\mathbf{L}_n)) + \beta\left\|\mathbf{L}_n\right\|_{F,\mathrm{off}}^2$.

Choosing $R(\cdot) = \left\|\cdot\right\|_{1,1}$ to promote the sparsity of the temporal variations, a formulation similar to the authors' prior work [32] is yielded, which is also solved via a primal-dual splitting method. In [28], several approaches for graph learning, including time-varying graphs, are proposed with applications in financial markets. The problem formulation for learning time-varying graphs in this paper is the same as that in the TVGLASSO method (2). However, here the structural constraints of the Laplacian matrices $\{\mathbf{L}_n|_{n=1}^N\}$ are incorporated into the formulation (as in (5)), and only causal batch data (past data samples) are used for graph learning. In [34], [35], a general framework for time-varying topology learning is introduced that can deal with online streaming data. Three types of models are studied in this paper, namely the time-varying Gaussian graphical model (TV-GGM), the time-varying smoothness-based model (TV-SBM), and the time-varying structural equation model (TV-SEM), where the first two apply to undirected graphical models, and the last one is utilized for directed topology identification. The methodology in this paper is based on a time-varying optimization framework proposed in [36]. Here, the general formulation for time-varying graph learning is as follows:

$$\mathbf{L}_t^\star = \underset{\mathbf{L}}{\mathrm{argmin}} \quad F(\mathbf{L};t) = f(\mathbf{L};t) + \lambda\, g(\mathbf{L};t), \tag{6}$$

where $\mathbf{L}_t^\star$ denotes the matrix representation for the graph model (e.g., the Laplacian matrix for an undirected graph). In this formulation, $f(\cdot)$ represents a smooth, strongly convex differentiable fidelity (similarity) measure, and $g(\cdot)$ denotes a potentially non-smooth convex regularization function. For instance, in the TV-GGM Gaussian graphical model, we have

$$\begin{aligned} f(\mathbf{L};t) &= -\log\det(\mathbf{L}) + \mathrm{tr}(\mathbf{L}\mathbf{\Sigma}_t), \\ g(\mathbf{L};t) &= g(\mathbf{L}) = i_{\Omega_\mathbf{L}}(\mathbf{L}) = \begin{cases} 0, & \mathbf{L}\in\Omega_\mathbf{L} \\ \infty, & \mathbf{L}\notin\Omega_\mathbf{L} \end{cases}, \end{aligned} \tag{7}$$

where $\Omega_\mathbf{L} = \{\mathbf{L} \succ \mathbf{0}\}$ denotes the set of positive definite matrices. In this problem, a solution is found via recursive prediction-correction steps. In the prediction step, a quadratic second-order approximation $\hat{F}(\mathbf{L};t+1)$ of the unobserved function $F(\mathbf{L};t+1)$ is minimized. In the correction step, the exact function $F(\mathbf{L};t+1)$ is optimized by exploiting the updated statistics of the data, with the new data sample $\mathbf{x}_t$ being received.

In addition to time-varying graph learning approaches for undirected structures, there have been efforts to learn time-varying directed topologies arising in structural equation models, including the works in [29] and [37]. Several other online approaches to graph topology identification have also been studied in the literature, e.g., [38]–[44].

Most of the existing approaches for time-varying graph learning are either tailored for offline (full-batch) data, where all the samples $\{1,\ldots,T\}$ are collected for topology identification [25]–[27], or they deal with online streaming data, where the graph topology is updated with every new data sample $\mathbf{x}_t$ being acquired [29], [35]. There is also a causal (batch) approach [28] that exploits all the past data frames (e.g., $1,\ldots,n$) for learning the graph at the current time frame (e.g., n). The offline (full-batch) and causal (batch) approaches suffer from delays and the need for large data storage, while the online approach is slow due to the high computational costs incurred by graph learning at every time stamp. This can make it impractical for real-time use unless some form of cache storage is utilized. Additionally, online approaches may struggle to efficiently capture the temporal consistency in the variations of the graph weights.

This challenge can be addressed by proposing a semi-online (mini-batch) approach in which the graph is updated using only the data samples from a single time frame. The length of the data frame T_n can then be adjusted based on how dynamic the desired graph topology is assumed to be, depending on the application.

The existing time-varying graph learning methods are also designed for Gaussian data, and they cannot efficiently deal with heavy-tailed data or data with outliers, which are very common in real-world applications, e.g., financial markets

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173

TABLE I: Summary of various benchmark methods for time-varying graph learning in comparison to the proposed approach. The objective function comprises a temporal consistency term, a graph regularization term, and a data fidelity terms. In this context, t represents samples at each time instant, n denotes the index of the frame (mini-batch), and F_n refers to the time indices within the n-th frame. The last column represents the worst-case arithmetic computational complexity of the algorithms in each iteration.

	Temporal Consistency	Graph Regularization + Data Fidelity	Constraints	Data	Complexity
TVGLASSO [26]	$\sum_{n=2}^{N} \|\mathbf{L}_n - \mathbf{L}_{n-1}\|_{p,q}^q$ $p, q \in \{1, 2, \infty\}$	$\sum_{n=1}^{N} \left(-T_n \log\det(\mathbf{L}_n) + \lambda \|\mathbf{L}_n\|_{1,\text{off}}\right)$ $+ \sum_{n=1}^{N} T_n \operatorname{tr}(\mathbf{\Sigma}_n \mathbf{L}_n)$	$\mathbf{L}_n \succ \mathbf{0}$	$\mathbf{X} = [\mathbf{X}_n]_{n=1}^{N}$ (Full-batch)	$\mathcal{O}(p^3)$
Kalofolias et al. [25]	$\sum_{n=2}^{N} \|\mathbf{W}_n - \mathbf{W}_{n-1}\|_F^2$	$\sum_{n=1}^{N} \left(-\alpha \mathbf{1}^\top \log(\mathbf{W}_n \mathbf{1}) + \beta \|\mathbf{W}_n\|_F^2\right)$ $+ \sum_{n=1}^{N} \operatorname{tr}(\mathbf{W}_n \mathbf{Z}_n^\top)$	$\mathbf{W}_n \geq \mathbf{0}$, $\mathbf{W}_n = \mathbf{W}_n^\top$, $\operatorname{diag}(\mathbf{W}_n) = \mathbf{0}$	$\mathbf{X} = [\mathbf{X}_n]_{n=1}^{N}$ (Full-batch)	$\mathcal{O}(p^2)$
Yamada et al. [32]	$\sum_{n=2}^{N} \|\mathbf{W}_n - \mathbf{W}_{n-1}\|_{1,1}$	$\sum_{n=1}^{N} \left(-\alpha \mathbf{1}^\top \log(\mathbf{W}_n \mathbf{1}) + \beta \|\mathbf{W}_n\|_F^2\right)$ $+ \sum_{n=1}^{N} \operatorname{tr}(\mathbf{W}_n \mathbf{Z}_n^\top)$	$\mathbf{W}_n \geq \mathbf{0}$, $\mathbf{W}_n = \mathbf{W}_n^\top$, $\operatorname{diag}(\mathbf{W}_n) = \mathbf{0}$	$\mathbf{X} = [\mathbf{X}_n]_{n=1}^{N}$ (Full-batch)	$\mathcal{O}(p^2)$
TV-GGM [34]	– – –	$-\log\det(\mathbf{L}_t) + \operatorname{tr}(\mathbf{L}_t \mathbf{\Sigma}_t)$	$\mathbf{L}_t \succ \mathbf{0}$	$\mathbf{x}_t$ (Online)	$\mathcal{O}(p^3)$
TV-SBM [35]	– – –	$-\alpha \mathbf{1}^\top \log(\mathbf{W}_t \mathbf{1}) + \beta \|\mathbf{W}_t\|_F^2$ $+ \operatorname{tr}\left((\operatorname{Diag}(\mathbf{W}_t \mathbf{1}) - \mathbf{W}_t)\mathbf{\Sigma}_t\right)$	$\mathbf{W}_t \geq \mathbf{0}$, $\mathbf{W}_t = \mathbf{W}_t^\top$, $\operatorname{diag}(\mathbf{W}_t) = \mathbf{0}$	$\mathbf{x}_t$ (Online)	$\mathcal{O}(p^3)$
Proposed	$\|\mathbf{w}_n - \mathbf{a} \odot \mathbf{w}_{n-1}\|_1$	$-\log\det^*(\mathcal{L}\mathbf{w}_n) + \beta \|\mathbf{w}_n\|_0$ $+ \frac{\nu+p}{T_n} \sum_{t \in F_n} \log\left(1 + \frac{\mathbf{x}_t^\top \mathcal{L}\mathbf{w}_n \mathbf{x}_t}{\nu}\right)$	$\mathbf{w}_n \geq \mathbf{0}$, $\eth \mathbf{w}_n = \mathbf{d}$, $\operatorname{rank}(\mathcal{L}\mathbf{w}_n) = p - k$	$\mathbf{X}_n = [\mathbf{x}_t, t \in F_n]$ (Mini-batch)	$\mathcal{O}(p^3)$

[45]. Hence, these methods cannot be applied to learning time-varying topologies of markets in finance. They cannot either learn graphs with specific spectral properties that can be used for clustering, e.g., k-component graphs, which are very applicable in unsupervised machine learning (data mining) [46].

Another issue with the current approaches is that they utilize a simple subtractive model for the graph variations, where the difference $\mathbf{w}_n - \mathbf{w}_{n-1}$ is assumed to be smooth. However, in many real cases, this model is not sufficient and a multiplicative factor $\mathbf{a}$ as $\mathbf{w}_n - \mathbf{a} \odot \mathbf{w}_{n-1}$ may be required to better model the variations.

Another drawback of the existing methods is that they rely on complete statistics of the data, while in many real-world applications (such as sensor networks), there may be missing entries in the data (due to sensor failure) or the data may be contaminated with noise (due to measurement errors). This issue has been addressed for static graph learning (e.g., [33], [47]) by adopting a joint signal and graph learning approach in which an additional signal denoising/imputation step is performed. However, this has not been well addressed for learning time-varying graphs.

In the following section, we address these issues by proposing a robust predictive approach to learning time-varying graphs with specific properties that apply to heavy-tailed data.

D. Contributions

Our contributions can be summarized as follows:

- We propose a probabilistic framework to model the signal and the weights of the graph in time-varying scenarios. Specifically, we utilize a non-negative vector autoregressive (VAR) model to capture the temporal variations in the weights of the graph. Our method is based on MAP estimation of the graph model in a semi-online approach in which the graph is only updated within frames, where the frame length can be adjusted to achieve the optimal balance between complexity and dynamics.
- We consider a heavy-tailed Student-t distribution for the signal characterized by the Laplacian matrix of a time-varying graph. This distribution can efficiently model data with outliers (e.g., financial data) and it can also handle Gaussian data by choosing the parameter ν large enough. Our method is also robust to measurement noise and missing data.
- We incorporate spectral and structural constraints into the problem of learning time-varying graphs. Our method can be used to learn k-component graphs applied to data clustering. We achieve this by imposing constraints on the node degrees and the rank of the Laplacian matrix.
- We propose an iterative method with proof of convergence, to solve the problem using the alternating direction method of multipliers (ADMM), where a majorization of the original function is optimized in each subproblem. Numerical results demonstrate that our proposed method outperforms some state-of-the-art algorithms for learning time-varying graph models, specifically from heavy-tailed data.

III. Proposed Approach

A. Model and Problem Formulation

In contrast to the deterministic approaches for time-varying graph learning, we propose a probabilistic framework to address the problem of learning a time-varying graph. To this end, we employ a non-negative VAR equation to characterize the variations in the edge weights of the graph as follows:

$$\mathbf{w}_n = \left(\mathbf{a} \odot \mathbf{w}_{n-1} + \boldsymbol{\epsilon}_n\right)_+, \quad n = 1, \ldots, N, \tag{8}$$

where the positive part function $(\cdot)_+$ is used to ensure the weights remain non-negative. Here, $\mathbf{a} \in \mathbb{R}_+^{p(p-1)/2}$ models the VAR coefficients vector assumed to have an exponential prior and $\boldsymbol{\epsilon}_n$ is a zero-mean temporally and spatially white innovation process with the Laplace distribution. The choice of the Laplace distribution for the innovations and the exponential distribution for $\mathbf{a}$ is justified for promoting sparsity in the graph weights. Then, we have:

$$\begin{aligned} p(\mathbf{a}) &= \lambda^{p(p-1)/2} \exp\left(-\lambda \mathbf{a}^\top \mathbf{1}\right), \quad \lambda > 0, \\ p(\boldsymbol{\epsilon}_n) &= \frac{1}{(2\sigma_\epsilon)^p} \exp\left(-\frac{\|\boldsymbol{\epsilon}_n\|_1}{\sigma_\epsilon}\right). \end{aligned} \tag{9}$$

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173

We also adopt a stochastic model for the signal, presuming that $\mathbf{x}_t$ for $t \in F_n$ follows a Laplacian heavy-tailed multivariate Student-t distribution, as follows:

$$p(\mathbf{x}_t|\mathbf{w}_n) \propto \det{}^*(\mathcal{L}\mathbf{w}_n)^{1/2} \left(1 + \frac{\mathbf{x}_t^\top \mathcal{L}\mathbf{w}_n \mathbf{x}_t}{\nu}\right)^{-(\nu+p)/2}, \quad t \in F_n, \quad \nu > 2. \tag{10}$$

Choosing the Student-t distribution for modeling heavy-tailed data, particularly in financial contexts, is extensively supported in the literature [45]. Now, suppose we have corrupted measurements of the signal, i.e., some samples are missing, and there is also some noise. Therefore, the measurements $\mathbf{y}_t$ are modeled as

$$\begin{aligned} &\mathbf{y}_t = \mathbf{m}_t \odot (\mathbf{x}_t + \mathbf{n}_t), \quad t \in \{1, \ldots, T\}, \\ &\mathbf{Y} = \mathbf{M} \odot (\mathbf{X} + \mathbf{N}), \end{aligned} \tag{11}$$

where $\mathbf{m}_t$ is a given sampling mask vector with binary elements (zeros correspond to missing samples), $\odot$ denotes the point-wise Hadamard product, and $\mathbf{n}_t$ is a zero-mean i.i.d. Gaussian noise vector with distribution $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$. We also have $\mathbf{Y} = [\mathbf{y}_t \,|_{t=1}^T]$, $\mathbf{M} = [\mathbf{m}_t \,|_{t=1}^T]$, and $\mathbf{N} = [\mathbf{n}_t \,|_{t=1}^T]$.

To estimate the time-varying graph weights $\mathbf{w}_n$, we employ a maximum a posteriori (MAP) estimation through a semi-online (mini-batch) approach, where only a single data frame is utilized for graph learning.

Let $\mathbf{Y}_n = [\mathbf{y}_t | \, t \in F_n] = \mathbf{M}_n \odot (\mathbf{X}_n + \mathbf{N}_n)$ denote the matrix of the corrupted signal samples (observations) at the time frame n. For inference of $\mathbf{w}_n$, the VAR model parameters, $\mathbf{a}$, and the original (clean) signal $\mathbf{x}_t$, in a semi-online fashion, data collection is limited to the n-th time frame, i.e., we only utilize $\mathbf{Y}_n$. In this scenario, we may need to await the availability of T_n data samples in time frame n. Nonetheless, this approach can also be adapted for online inference by setting $T_n = 1$. The problem can then be expressed as follows:

$$\begin{aligned} \min_{\mathbf{w}_n \geq \mathbf{0}, \mathbf{a} \geq \mathbf{0}, \mathbf{X}_n} \quad & -\log p(\mathbf{w}_n, \mathbf{a}, \mathbf{X}_n | \mathbf{w}_{n-1}, \mathbf{Y}_n, \mathbf{M}_n) \\ \text{s.t.} \quad & \mathbf{w}_n \in \Omega_{\mathbf{w}} \end{aligned} \tag{12}$$

where $-\log p(\mathbf{w}_n, \mathbf{a}, \mathbf{X}_n | \mathbf{w}_{n-1}, \mathbf{Y}_n, \mathbf{M}_n) = -\log p(\mathbf{X}_n|\mathbf{w}_n) - \log p(\mathbf{Y}_n|\mathbf{X}_n, \mathbf{M}_n) - \log p(\mathbf{w}_n|\mathbf{w}_{n-1}, \mathbf{a}) - \log p(\mathbf{a}) + \text{const}$ and $\Omega_{\mathbf{w}}$ represents the set of equality constraints that define the feasible region of the desired graph weights. In particular, we assume that the underlying graph structure is k connected and that the degrees of the vertices are fixed and equal to a constant vector $\mathbf{d}$, i.e., $\Omega_{\mathbf{w}} = \{\mathbf{w} | \, \text{rank}(\mathcal{L}\mathbf{w}) = p - k, \mathfrak{d}\mathbf{w} = \mathbf{d}\}$, where $\mathfrak{d}$ represents the degree operator, mapping the vector of edge weights to the vertex degrees [48]. The rank constraint ensures that the graph is k-component, as the number of zero eigenvalues of the Laplacian matrix of an undirected graph is equal to the number of disjoint components of the graph. Hence, for a graph with k components, the Laplacian matrix is of rank $p - k$. This constraint helps learn multi-component graphs for efficient representation of data with inherent cluster structure, where each graph component represents one data cluster. This also provides more flexibility (degree of freedom) in graph modeling, since our model can also be used for connected graph learning by setting $k = 1$. The degree constraint $\mathfrak{d}\mathbf{w} = \mathbf{d}$ also controls the distribution of the edge weights of the graph and helps avoid isolated nodes. Assume the (non-negative) weights of edges from node i represent the probabilities with which node i is connected to other nodes. In this case the degree constraint with $\mathbf{d} = \mathbf{1}$ ensures the sum of these probabilities is equal to one.

Proposition 1. *Let $\mathbf{w}_{n-1}$ in (12) be replaced by an estimate of the graph weights from the previous time frame, denoted as $\hat{\mathbf{w}}_{n-1}$. By expanding the posterior probability for MAP estimation and simplifying, we obtain the following formulation for learning the time-varying graph:*

$$\begin{aligned} \min_{\mathbf{w}_n \geq \mathbf{0}, \mathbf{a} \geq \mathbf{0}, \mathbf{X}_n} \quad & \tfrac{1}{T_n \sigma_n^2} \left\| \mathbf{Y}_n - \mathbf{M}_n \odot \mathbf{X}_n \right\|_F^2 - \log \det{}^*(\mathcal{L}\mathbf{w}_n) \\ & + \tfrac{\nu+p}{T_n} \textstyle\sum_{t \in F_n} \log \left(1 + \frac{\mathbf{x}_t^\top \mathcal{L}\mathbf{w}_n \mathbf{x}_t}{\nu}\right) \\ & + \alpha \left\| \mathbf{w}_n - \mathbf{a} \odot \hat{\mathbf{w}}_{n-1} \right\|_1 + \beta \left\| \mathbf{w}_n \right\|_0 + \gamma \mathbf{a}^\top \mathbf{1} \\ \text{s.t.} \quad & \mathbf{w}_n \in \Omega_{\mathbf{w}} \end{aligned} \tag{13}$$

where $\alpha = \frac{2}{T_n \sigma_\epsilon}$, $\beta = \frac{2 \log \sigma_\epsilon}{T_n}$, and $\gamma = \frac{2\lambda}{T_n}$.

Proof. See appendix A. □

B. Solution

The non-convex formulation of problem (13), coupled with the interleaved equality constraints on $\mathbf{w}_n$, renders it challenging to solve. Nevertheless, taking advantage of splitting techniques in convex optimization, particularly ADMM [49], a promising approach to address this problem can be devised. We begin by reformulating the problem, introducing the auxiliary variables $\mathbf{L}_n = \mathcal{L}\mathbf{w}_n$ and $\mathbf{u}_n = \mathbf{w}_n - \mathbf{a} \odot \hat{\mathbf{w}}_{n-1}$. We further incorporate an additional penalty function $\text{tr}\left(\mathcal{L}\mathbf{w}_n \mathbf{V}_n \mathbf{V}_n^\top\right)$ with $\mathbf{V}_n \in \mathbb{R}^{p \times k}$, $\mathbf{V}_n^\top \mathbf{V}_n = \mathbf{I}$ to more effectively control the rank of $\mathcal{L}\mathbf{w}_n$. Thus, the problem formulation becomes:

$$\begin{aligned} \min_{\substack{\mathbf{w}_n \geq \mathbf{0}, \mathbf{a} \geq \mathbf{0}, \\ \mathbf{X}_n, \mathbf{u}_n, \mathbf{L}_n, \mathbf{V}_n}} \quad & f(\mathbf{w}_n, \mathbf{X}_n, \mathbf{a}, \mathbf{u}_n, \mathbf{L}_n, \mathbf{V}_n) \triangleq \\ & -\log \det{}^*(\mathbf{L}_n) + \alpha \left\| \mathbf{u}_n \right\|_1 + \beta \left\| \mathbf{w}_n \right\|_0 \\ & + \tfrac{\nu+p}{T_n} \textstyle\sum_{t \in F_n} \log \left(1 + \frac{\mathbf{x}_t^\top \mathcal{L}\mathbf{w}_n \mathbf{x}_t}{\nu}\right) \\ & + \tfrac{1}{T_n \sigma_n^2} \left\| \mathbf{Y}_n - \mathbf{M}_n \odot \mathbf{X}_n \right\|_F^2 + \gamma \mathbf{a}^\top \mathbf{1} \\ & + \eta \, \text{tr}\left(\mathcal{L}\mathbf{w}_n \mathbf{V}_n \mathbf{V}_n^\top\right) \\ \text{s.t.} \quad & \mathbf{L}_n = \mathcal{L}\mathbf{w}_n, \;\; \mathbf{u}_n = \mathbf{w}_n - \mathbf{a} \odot \hat{\mathbf{w}}_{n-1}, \\ & \mathfrak{d}\mathbf{w}_n = \mathbf{d}, \;\; \text{rank}(\mathbf{L}_n) = p - k, \;\; \mathbf{V}_n^\top \mathbf{V}_n = \mathbf{I}. \end{aligned} \tag{14}$$

Hence, the augmented Lagrangian for this problem yields as follows:

$$\begin{aligned} & L_\rho(\mathbf{w}_n, \mathbf{X}_n, \mathbf{a}, \mathbf{u}_n, \mathbf{L}_n, \mathbf{V}_n, \mathbf{\Phi}_n, \boldsymbol{\mu}_n, \mathbf{z}_n) = \\ & \quad f(\mathbf{w}_n, \mathbf{X}_n, \mathbf{a}, \mathbf{u}_n, \mathbf{L}_n, \mathbf{V}_n) \\ & \quad + \frac{\rho}{2} \left\| \mathcal{L}\mathbf{w}_n - \mathbf{L}_n \right\|_F^2 + \langle \mathcal{L}\mathbf{w}_n - \mathbf{L}_n, \mathbf{\Phi}_n \rangle \\ & \quad + \frac{\rho}{2} \left\| \mathbf{u}_n - \mathbf{w}_n + \mathbf{a} \odot \hat{\mathbf{w}}_{n-1} \right\|^2 + \langle \mathbf{u}_n - \mathbf{w}_n + \mathbf{a} \odot \hat{\mathbf{w}}_{n-1}, \boldsymbol{\mu}_n \rangle \\ & \quad + \frac{\rho}{2} \left\| \mathfrak{d}\mathbf{w}_n - \mathbf{d} \right\|^2 + \langle \mathfrak{d}\mathbf{w}_n - \mathbf{d}, \mathbf{z}_n \rangle. \end{aligned} \tag{15}$$

Now, employing ADMM, we derive an iterative solution consisting of six update steps for the primal variables $\mathbf{w}_n$, $\mathbf{X}_n$, $\mathbf{a}$, $\mathbf{u}_n$, $\mathbf{L}_n$ and $\mathbf{V}_n$, along with three update steps for the dual variables $\mathbf{\Phi}_n$, $\boldsymbol{\mu}_n$ and $\mathbf{z}_n$.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173

$\mathbf{L}_n$-update step

The subproblem associated with the update step of $\mathbf{L}$ possesses a closed-form solution given by:

$$\begin{aligned}\mathbf{L}_n^{l+1} &= \underset{\mathbf{L}_n\succeq\mathbf{0},\,\mathrm{rank}(\mathbf{L}_n)=p-k}{\operatorname{argmin}} -\log\det{}^*(\mathbf{L}_n)+\frac{\rho}{2}\left\|\mathbf{L}_n-\mathcal{L}\mathbf{w}_n^l-\frac{1}{\rho}\mathbf{\Phi}_n^l\right\|_F^2\\ &= \frac{1}{2}\mathbf{U}^l\left(\mathbf{\Gamma}^l+\left(\mathbf{\Gamma}^{l2}+\frac{4}{\rho}\mathbf{I}\right)^{1/2}\right)\mathbf{U}^{l\top}.\end{aligned} \tag{16}$$

Here, $\mathbf{\Gamma}^l$ is a diagonal matrix comprising only the largest $p-k$ eigenvalues of $\mathcal{L}\mathbf{w}_n^l+\mathbf{\Phi}_n^l/\rho$, with their corresponding eigenvectors contained in $\mathbf{U}^l\in\mathbb{R}^{p\times(p-k)}$.

$\mathbf{w}_n$-update step

The subproblem related to the update step of $\mathbf{w}$ is expressed as follows:

$$\begin{aligned}\mathbf{w}_n^{l+1} = \underset{\mathbf{w}_n\geq\mathbf{0}}{\operatorname{argmin}}\ &\frac{p+\nu}{T_n}\sum_{t\in F_n}\log\left(1+\frac{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n\,\mathbf{x}_t^l}{\nu}\right)+\beta\left\|\mathbf{w}_n\right\|_0\\ &+\frac{\rho}{2}\left\|\mathcal{L}\mathbf{w}_n-\mathbf{L}_n^l+\frac{1}{\rho}\mathbf{\Phi}_n^l\right\|_F^2\\ &+\frac{\rho}{2}\left\|\mathbf{u}_n^l-\mathbf{w}_n+\mathbf{a}^l\odot\hat{\mathbf{w}}_{n-1}+\frac{1}{\rho}\boldsymbol{\mu}_n^l\right\|^2\\ &+\frac{\rho}{2}\left\|\mathfrak{d}\mathbf{w}_n-\mathbf{d}+\frac{1}{\rho}\mathbf{z}_n^l\right\|^2+\eta\,\mathrm{tr}\left(\mathcal{L}\mathbf{w}_n\mathbf{V}_n^l\mathbf{V}_n^{l\top}\right).\end{aligned} \tag{17}$$

To address this challenging problem, we employ the majorization minimization (MM) technique [50] to minimize a surrogate majorization function of the original cost. First, using the inequality $\log(x)\leq x-1,\ \forall x>0$, we obtain:

$$\begin{aligned}\log\left(1+\frac{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n\,\mathbf{x}_t^l}{\nu}\right) &\leq \log\left(1+\frac{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^l\,\mathbf{x}_t^l}{\nu}\right)\\ &\quad+\frac{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n\,\mathbf{x}_t^l+\nu}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^l\,\mathbf{x}_t^l+\nu}-1 \qquad (18)\\ &= \left\langle\mathbf{w}_n,\mathcal{L}^*\left(\frac{\mathbf{x}_t^l\mathbf{x}_t^{l\top}}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^l\,\mathbf{x}_t^l+\nu}\right)\right\rangle+\\ &\quad c_0(\mathbf{w}_n^l).\end{aligned}$$

Here, $\mathbf{w}_n^l$ represents a fixed point, selected as the solution from the previous iteration and $c_0(\cdot)$ is a constant function. We can also define a majorization function for the term $\|\mathcal{L}\mathbf{w}_n\|_F^2+\|\mathfrak{d}\mathbf{w}_n\|^2=\mathbf{w}_n^\top\mathbf{H}\mathbf{w}_n$ as

$$\begin{aligned}\mathbf{w}_n^\top\mathbf{H}\mathbf{w}_n &\leq \mathbf{w}_n^\top\mathbf{H}\mathbf{w}_n+(\mathbf{w}_n-\mathbf{w}_n^l)^\top(\zeta\mathbf{I}-\mathbf{H})(\mathbf{w}_n-\mathbf{w}_n^l)\\ &= \zeta\left\|\mathbf{w}_n-\mathbf{w}_n^l\right\|^2+2\langle\mathbf{w}_n,\mathbf{H}\mathbf{w}_n^l\rangle+c_1(\mathbf{w}_n^l),\end{aligned} \tag{19}$$

where $\mathbf{H}=\mathcal{L}^*\mathcal{L}+\mathfrak{d}^*\mathfrak{d}$ and $\zeta\geq\lambda_{\max}(\mathbf{H})=4p-2$ [19]. Using the proposed majorization functions (upper-bounds) in (18) and (19) with $\zeta=4p-2$, the update step for $\mathbf{w}_n$ would be simplified as follows:

$$\begin{aligned}\mathbf{w}_n^{l+1} &= \underset{\mathbf{w}_n\geq\mathbf{0}}{\operatorname{argmin}}\ \frac{\rho(4p-1)}{2}\left\|\mathbf{w}_n-\mathbf{c}_{\mathbf{w}}^l\right\|^2+\beta\left\|\mathbf{w}_n\right\|_0\\ &= \mathbb{1}(\mathbf{c}_{\mathbf{w}}^l>\mathbf{c}_{th})\odot\mathbf{c}_{\mathbf{w}}^l,\end{aligned} \tag{20}$$

where $\mathbf{c}_{th}=\sqrt{\frac{2\beta}{\rho(4p-1)}}\mathbf{1}$ and

$$\begin{aligned}\mathbf{c}_{\mathbf{w}}^l &= \left(1-\tfrac{\rho}{\rho(4p-1)}\right)\mathbf{w}_n^l-\frac{1}{\rho(4p-1)}\left(\mathbf{a}_{\mathbf{w}}^l+\mathbf{b}_{\mathbf{w}}^l\right),\\ \mathbf{a}_{\mathbf{w}}^l &= \mathcal{L}^*\left(\tilde{\mathbf{S}}^l+\mathbf{\Phi}_n^l+\rho\left(\mathcal{L}\mathbf{w}_n^l-\mathbf{L}_n^{l+1}\right)+\eta\mathbf{V}_n^l\mathbf{V}_n^{l\top}\right),\\ \mathbf{b}_{\mathbf{w}}^l &= -\boldsymbol{\mu}_n^l-\rho\left(\mathbf{u}_n^l+\mathbf{a}^l\odot\hat{\mathbf{w}}_{n-1}\right)+\mathfrak{d}^*\left(\mathbf{z}_n^l-\rho\left(\mathbf{d}-\mathfrak{d}\mathbf{w}_n^l\right)\right),\\ \tilde{\mathbf{S}}^l &= \tfrac{p+\nu}{T_n}\textstyle\sum_{t\in F_n}\frac{\mathbf{x}_t^l\mathbf{x}_t^{l\top}}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^l\mathbf{x}_t^l+\nu}.\end{aligned} \tag{21}$$

In the equation above, $\mathcal{L}^*$ and $\mathfrak{d}^*$ represent the adjoints of the Laplacian and the degree operators, as defined in [19].

$\mathbf{u}_n$-update step

The subproblem associated with to the update step of $\mathbf{u}_n$ admits a closed-form solution given by:

$$\begin{aligned}\mathbf{u}_n^{l+1} &= \underset{\mathbf{u}_n}{\operatorname{argmin}}\ \frac{\rho}{2}\left\|\mathbf{u}_n-\mathbf{w}_n^{l+1}+\mathbf{a}^l\odot\hat{\mathbf{w}}_{n-1}+\frac{1}{\rho}\boldsymbol{\mu}_n^l\right\|^2\\ &\qquad+\alpha\left\|\mathbf{u}_n\right\|_1 \qquad (22)\\ &= \mathcal{S}_{\frac{\alpha}{\rho}}\left(\mathbf{w}_n^{l+1}-\mathbf{a}^l\odot\hat{\mathbf{w}}_{n-1}-\frac{1}{\rho}\boldsymbol{\mu}_n^l\right),\end{aligned}$$

where $\mathcal{S}$ denotes the soft-thresholding operator [51].

$\mathbf{X}_n$-update step

The update step for $\mathbf{X}_n$ is obtained by solving the following problem:

$$\begin{aligned}\mathbf{X}_n^{l+1} &= \{\mathbf{x}_t^{l+1}|_{t\in F_n}\} = \underset{\{\mathbf{x}_t|_{t\in F_n}\}}{\operatorname{argmin}}\ \sum_{t\in F_n}f_{\mathbf{x}_t}(\mathbf{x}_t)\\ f_{\mathbf{x}_t}(\mathbf{x}_t) &= \frac{1}{T_n\sigma_n^2}\left\|\mathbf{y}_t-\mathbf{m}_t\odot\mathbf{x}_t\right\|^2\\ &\qquad+\frac{p+\nu}{T_n}\log\left(1+\frac{\mathbf{x}_t^\top\mathcal{L}\mathbf{w}_n^{l+1}\,\mathbf{x}_t}{\nu}\right).\end{aligned} \tag{23}$$

This can be decomposed into smaller problems for each $\mathbf{x}_t$ as follows:

$$\mathbf{x}_t^{l+1}=\underset{\mathbf{x}_t}{\operatorname{argmin}}\ f_{\mathbf{x}_t}(\mathbf{x}_t),\quad t\in F_n. \tag{24}$$

To find a closed-form solution to this problem, we replace $f_{\mathbf{x}_t}(\mathbf{x}_t)$ with a majorization function as proposed by the following proposition.

Proposition 2. *Let* $\tau\geq\frac{1}{\sigma_n^2}+\frac{p+\nu}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu}\lambda_{\max}(\mathcal{L}\mathbf{w}_n^{l+1})$. *Define the function*

$$f_{\mathbf{x}_t}^S(\mathbf{x}_t,\mathbf{x}_0)=\frac{\tau}{T_n}\left\|\mathbf{x}_t-\mathbf{x}_0+\frac{\mathbf{Q}_t\mathbf{x}_0-\mathbf{c}_t}{\tau}\right\|^2+C(\mathbf{x}_0), \tag{25}$$

where

$$\begin{aligned}\mathbf{Q}_t &= \frac{1}{\sigma_n^2}\mathrm{Diag}(\mathbf{m}_t)+\frac{p+\nu}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu}\mathcal{L}\mathbf{w}_n^{l+1},\\ \mathbf{c}_t &= \frac{1}{\sigma_n^2}\mathbf{y}_t.\end{aligned} \tag{26}$$

Here, $\mathbf{x}_0$ *and* $C(\mathbf{x}_0)$ *are constants. Then,* $f_{\mathbf{x}_t}^S(\mathbf{x}_t,\mathbf{x}_0)$ *serves as a majorization function for* $f_{\mathbf{x}_t}(\mathbf{x}_t)$*, satisfying the inequality* $f_{\mathbf{x}_t}(\mathbf{x}_t)\leq f_{\mathbf{x}_t}^S(\mathbf{x}_t,\mathbf{x}_0),\ \forall\mathbf{x}_0$.

Proof. See appendix B. □

By applying this proposition with $\mathbf{x}_0 = \mathbf{x}_t^l$ in the context of a majorization-minimization framework, we formulate and solve the following problem for the update step of $\mathbf{x}_t$:

$$\begin{aligned}\mathbf{x}_t^{l+1} &= \underset{\mathbf{x}_t}{\operatorname{argmin}}\, f_{\mathbf{x}_t}^S(\mathbf{x}_t, \mathbf{x}_t^l) \\ &= \mathbf{x}_t^l - \frac{1}{\tau}\left(\mathbf{Q}_t\mathbf{x}_t^l - \frac{1}{\sigma_n^2}\mathbf{y}_t\right).\end{aligned} \tag{27}$$

$\mathbf{a}$-*update step*

The VAR parameters vector $\mathbf{a}$ can also be updated using the following closed-form solution:

$$\begin{aligned}\mathbf{a}^{l+1} &= \underset{\mathbf{a}\geq\mathbf{0}}{\operatorname{argmin}}\, \frac{\rho}{2}\left\|\mathbf{a}\odot\hat{\mathbf{w}}_{n-1} - \mathbf{f}^l\right\|^2 + \gamma\mathbf{a}^\top\mathbf{1} \\ &= \mathcal{S}_{\frac{\gamma}{\rho\hat{\mathbf{w}}_{n-1}^{\circ 2}}}\left((\mathbf{f}^l)_+ \oslash \hat{\mathbf{w}}_{n-1}\right)\odot\mathbb{1}(\hat{\mathbf{w}}_{n-1}>0),\end{aligned} \tag{28}$$

where $\mathbf{f}^l = \mathbf{w}_n^{l+1} - \mathbf{u}_n^{l+1} - \frac{1}{\rho}\boldsymbol{\mu}_n^l$.

$\mathbf{V}_n$-*update step*

Next, we have the update formula for $\mathbf{V}_n$ as follows:

$$\mathbf{V}_n^{l+1} = \underset{\mathbf{V}_n\in\mathbb{R}^{p\times k},\mathbf{V}_n^\top\mathbf{V}_n=\mathbf{I}}{\operatorname{argmin}} \operatorname{tr}\left(\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{V}_n\mathbf{V}_n^\top\right) = \mathbf{Q}_n^{l+1}[:,1:k]. \tag{29}$$

Here, $\mathbf{Q}_n^{l+1}[:,1:k]$ denotes the set of eigenvectors of $\mathcal{L}\mathbf{w}_n^{l+1}$ corresponding to the first k eigenvalues, sorted in ascending order.

Update step for the dual variables

Finally, we have the update step for the dual variables as follows:

$$\begin{aligned}\boldsymbol{\Phi}_n^{l+1} &= \boldsymbol{\Phi}_n^l + \rho\left(\mathcal{L}\mathbf{w}_n^{l+1} - \mathbf{L}_n^{l+1}\right), \\ \boldsymbol{\mu}_n^{l+1} &= \boldsymbol{\mu}_n^l + \rho\left(\mathbf{u}_n^{l+1} - \mathbf{w}_n^{l+1} + \mathbf{a}^{l+1}\odot\hat{\mathbf{w}}_{n-1}\right), \\ \mathbf{z}_n^{l+1} &= \mathbf{z}_n^l + \rho\left(\mathfrak{d}\mathbf{w}_n^{l+1} - \mathbf{d}\right).\end{aligned} \tag{30}$$

The proposed method, called k-component time-varying graph learning (*k-TVGL*), is summarized in Algorithm 1, with Theorem 1 establishing its convergence. The code for this algorithm is available as an open source repository at https://github.com/javaheriamirhossein/k-tvgraph.

Theorem 1. *The sequence of the augmented Lagrangian* $\left\{L_\rho\left(\mathbf{L}_n^l, \mathbf{w}_n^l, \mathbf{u}_n^l, \mathbf{X}_n^l, \mathbf{a}^l, \mathbf{V}_n^l, \boldsymbol{\Phi}_n^l, \boldsymbol{\mu}_n^l, \mathbf{z}_n^l\right)\right\}$ *generated by Algorithm 1 converges for any sufficiently large* ρ. *At the limit point, the equality constraints* $\mathbf{L}_n = \mathcal{L}\mathbf{w}_n$, $\mathbf{u}_n = \mathbf{w}_n - \mathbf{a}\odot\hat{\mathbf{w}}_{n-1}$, *and* $\mathfrak{d}\mathbf{w}_n = \mathbf{d}$ *are also satisfied, i.e.,*

$$\begin{aligned}&\lim_{l\to+\infty}\left\|\mathcal{L}\mathbf{w}_n^l - \mathbf{L}_n^l\right\|_F = 0 \\ &\lim_{l\to+\infty}\left\|\mathfrak{d}\mathbf{w}_n^l - \mathbf{d}\right\| = 0 \\ &\lim_{l\to+\infty}\left\|\mathbf{u}_n^l - \mathbf{w}_n^l + \mathbf{a}^l\odot\hat{\mathbf{w}}_{n-1}\right\| = 0.\end{aligned} \tag{31}$$

Proof. See the supplementary notes. □

Algorithm 1 Proposed *k-TVGL* algorithm for k-component time-varying graph learning from heavy-tailed data

Input: The observation matrix $\mathbf{Y}_n = [\mathbf{y}_t | t \in F_n]$, the sampling mask $\mathbf{M}_n$ at time frame n, and $\hat{\mathbf{w}}_{n-1}$
Parameters: k, $\mathbf{d}$, ν, σ_ϵ, γ, ρ
Output: The signal and the estimated graph weights at time frame n, i.e., $\mathbf{X}_n^l$ and $\mathbf{w}_n^l$
Initialization: $\mathbf{w}_n^0 = \hat{\mathbf{w}}_{n-1}$, $\boldsymbol{\Phi}_n^0 = \mathbf{0}$, $\boldsymbol{\mu}_n^0 = \mathbf{0}$, $\mathbf{z}_n^0 = \mathbf{0}$, $\mathbf{X}_n^0 = \mathbf{Y}_n$, $\mathbf{a}^0$
Set $l = 0$
repeat
  Update $\mathbf{L}_n^{l+1}$ using (16).
  Update $\mathbf{w}_n^{l+1}$ via (20).
  Update $\mathbf{u}_n^{l+1}$ via (22).
  Update $\mathbf{x}_t^{l+1}$ for $t \in F_n$ via (27).
  Update $\mathbf{a}^{l+1}$ using (28).
  Update $\mathbf{V}_n^{l+1}$ using (29).
  Update the dual variables via (30).
  Set $l \leftarrow l + 1$.
until a stopping criterion is satisfied
Set $\hat{\mathbf{w}}_n = \mathbf{w}_n^l$.

C. Computational Complexity

The update step for $\mathbf{w}_n$ involves the computation of $\tilde{\mathbf{S}}$, which has a complexity of $\mathcal{O}(T_n p^2)$. Given this, the complexity of the closed-form solution in (20) would be $\mathcal{O}(p^2)$. This also holds for the update of $\mathbf{u}_n$ via (22). However, the update steps in (16) and (29) require eigenvalue decomposition of $p\times p$ matrices, which is generally $\mathcal{O}(p^3)$ complex. Hence, the overall complexity of the proposed algorithm is $\mathcal{O}(p^3 + T_n p^2)$. While this may indicate that the proposed method may not be scalable to very large graphs, it is intrinsic to every graph-based clustering method that deals with the eigenvectors of the graph Laplacian.

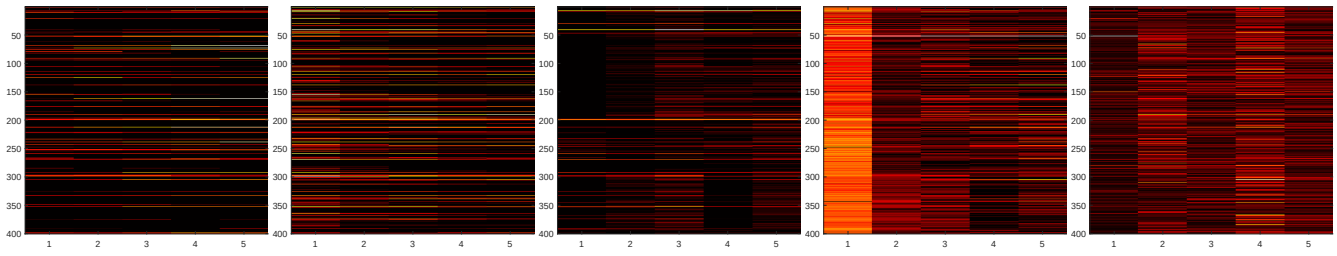
IV. Numerical Results

In this section, we present the numerical results of the proposed algorithm, comparing it to several state-of-the-art methods for time-varying graph learning across different scenarios. First, we evaluate the performance of our algorithm in learning a time-varying graph topology through a simulated experiment using synthetic data. Subsequently, we explore the application of this methodology in financial market analysis, focusing on data clustering and portfolio design. The results are detailed in the following two subsections.

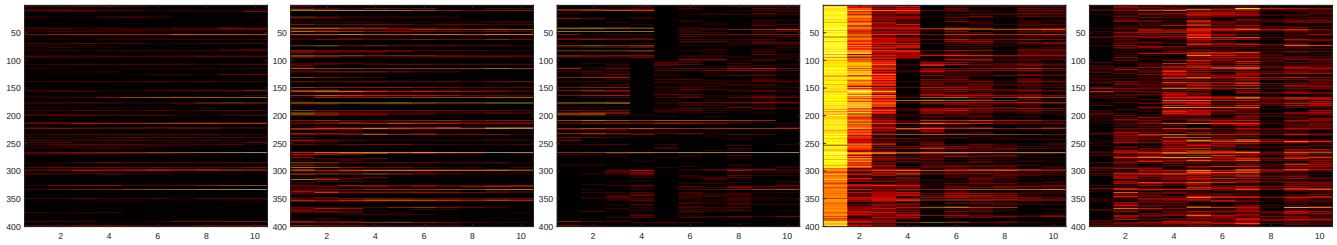
A. Synthetic Data

For synthetic data generation, we consider $p = 100$ and $T = 1000$. We divide the T time-stamps into equal frames (windows) of length T_n (with no overlap), where the graph is assumed to be constant during each time frame. Let F_n denote the time indices in frame n. Random samples of the signal in each frame are generated via $\mathbf{x}_t = \left(\mathbf{L}_n^\dagger\right)^{1/2}\boldsymbol{\eta}_t$, $\boldsymbol{\eta}_t \sim \mathrm{St}(\mathbf{0}, \mathbf{I}, \nu)$, $t \in F_n$, where $\mathbf{L}_n^\dagger$ represents the pseudo-inverse of the Laplacian matrix at time frame n and St denotes the

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173



(a) True weights (b) Proposed Method (c) Kalofolias et al. [25] (d) Cardoso et al. [28] (e) TV-GGM [35]



(f) True weights (g) Proposed Method (h) Kalofolias et al. [25] (i) Cardoso et al. [28] (j) TV-GGM [35]

Fig. 3: Visualization of the edge weights of the learned graphs at different time frames, with frames of length $T_n = 200$ (top) and $T_n = 100$ (bottom). The horizontal axis shows the time frame index and the vertical axis represents the edge index of the graph.

TABLE II: Average single-frame performance of the graph learning methods for time-varying topology identification at different sampling rates SR and fixed noise level $\sigma_n = 0$ (the values represent the mean $\pm$ standard deviation of the average performance over all data frames for 20 random trials with frame length $T_n = 200$).

	SR $= 1$			SR $= 0.8$			SR $= 0.6$		
	F-score ↑	RelErr ↓	Time (s) ↓	F-score ↑	RelErr ↓	Time (s) ↓	F-score ↑	RelErr ↓	Time (s) ↓
Kalofolias et al. [25]	0.61 ± 0.02	0.38 ± 0.01	**0.02 ± 0.00**	0.55 ± 0.01	0.43 ± 0.01	**0.02 ± 0.00**	0.49 ± 0.01	0.45 ± 0.01	**0.02 ± 0.00**
Cardoso et al. [28]	0.43 ± 0.00	0.34 ± 0.00	0.21 ± 0.10	0.36 ± 0.01	0.36 ± 0.04	0.24 ± 0.14	0.30 ± 0.00	0.37 ± 0.02	0.17 ± 0.02
TV-SBM [35]	0.55 ± 0.01	0.39 ± 0.02	61.98 ± 0.27	0.51 ± 0.01	0.40 ± 0.02	61.82 ± 0.19	0.43 ± 0.01	0.43 ± 0.01	61.91 ± 0.18
TV-GGM [34]	0.31 ± 0.01	0.37 ± 0.00	0.71 ± 0.01	0.30 ± 0.01	0.38 ± 0.01	0.72 ± 0.03	0.28 ± 0.01	0.39 ± 0.02	0.72 ± 0.02
Saboksayr et al. [41]	0.37 ± 0.01	0.46 ± 0.03	0.25 ± 0.01	0.34 ± 0.01	0.48 ± 0.01	0.25 ± 0.02	0.32 ± 0.00	0.49 ± 0.02	0.24 ± 0.01
Proposed (Alg. 1)	**0.69 ± 0.01**	**0.31 ± 0.01**	0.18 ± 0.00	**0.65 ± 0.01**	**0.31 ± 0.00**	0.18 ± 0.00	**0.56 ± 0.00**	**0.32 ± 0.00**	0.19 ± 0.02

TABLE III: Average single-frame performance of the graph learning methods for time-varying topology identification at different noise levels σ_n and fixed sampling rate SR $= 1$ (the values represent the mean $\pm$ standard deviation of the average performance over all data frames for 20 random trials with frame length $T_n = 200$).

	$\sigma_n = 0.3$			$\sigma_n = 0.5$			$\sigma_n = 1$		
	F-score ↑	RelErr ↓	Time (s) ↓	F-score ↑	RelErr ↓	Time (s) ↓	F-score ↑	RelErr ↓	Time (s) ↓
Kalofolias et al. [25]	0.59 ± 0.01	0.37 ± 0.02	**0.02 ± 0.00**	0.53 ± 0.01	0.37 ± 0.00	**0.02 ± 0.01**	0.45 ± 0.00	0.38 ± 0.01	**0.02 ± 0.01**
Cardoso et al. [28]	0.43 ± 0.00	0.34 ± 0.02	0.16 ± 0.02	0.43 ± 0.01	0.34 ± 0.01	0.19 ± 0.05	0.40 ± 0.01	0.37 ± 0.02	0.22 ± 0.01
TV-SBM [35]	0.54 ± 0.01	0.38 ± 0.01	61.94 ± 0.09	0.52 ± 0.01	0.39 ± 0.01	61.96 ± 0.10	0.42 ± 0.01	0.43 ± 0.01	61.97 ± 0.13
TV-GGM [34]	0.29 ± 0.00	0.38 ± 0.04	0.72 ± 0.03	0.29 ± 0.01	0.49 ± 0.15	0.72 ± 0.02	0.27 ± 0.01	0.70 ± 0.24	0.72 ± 0.04
Saboksayr et al. [41]	0.35 ± 0.00	0.44 ± 0.02	0.25 ± 0.00	0.32 ± 0.01	0.45 ± 0.01	0.25 ± 0.01	0.30 ± 0.00	0.47 ± 0.02	0.25 ± 0.01
Proposed (Alg. 1)	**0.65 ± 0.01**	**0.32 ± 0.02**	0.18 ± 0.00	**0.58 ± 0.01**	**0.33 ± 0.01**	0.18 ± 0.01	**0.47 ± 0.01**	**0.35 ± 0.02**	0.18 ± 0.01

Student-t distribution with zero mean and identity covariance matrix. To model the temporal variations of the graph weights, we use the equation $\mathbf{w}_n = (\mathbf{a} \odot \mathbf{w}_{n-1} + \boldsymbol{\epsilon}_n)_+, \quad n \in \{1, \ldots, N\}$. Here, $\mathbf{a}$ is sampled from an exponential distribution and $\boldsymbol{\epsilon}_n$ is sampled from a normal distribution, where N denotes the number of time frames. The initial values for the graph weights $\mathbf{w}_0$ (and subsequently the Laplacian matrix $\mathbf{L}_0$) are sampled from the Stochastic Block Model, where the nodes are partitioned into $K = 4$ clusters (blocks) with random intra-cluster and inter-cluster edges (with probabilities $p_{\text{intra}} = 0.7$ and $p_{\text{inter}} = 0.3$ respectively).

We construct the original data matrix $\mathbf{X}$ by concatenating the vectors $\mathbf{x}_t$ column-wise, covering the range from $t = 1$ to $t = T$. Following this, we normalize the data matrix such that each row is centered and scaled by its standard deviation. Next, we create a random binary sampling matrix $\mathbf{M}$ defined by the sampling rate parameter SR, along with the observation noise matrix $\mathbf{N}$, which consists of i.i.d. Gaussian random entries with zero mean and variance σ_n^2. The observation matrix is then formed as $\mathbf{Y} = \mathbf{M} \odot (\mathbf{X} + \mathbf{N})$.

We then introduce the matrices $\mathbf{Y}$ and $\mathbf{M}$ as inputs to the time-varying graph learning algorithms. We compare our method with several benchmark algorithms, including the TV-GGM [34] and the TV-SBM [35] methods, for online time-varying graph learning under Gaussian graphical and smoothness-based models. Additionally, we include the time-varying graph learning method in [28], the online graph learning algorithm by Saboksayr et al. [41], and the time-varying version of the GSP Toolbox[1] graph learning method by Kalofolias et al. [25] in our comparison.

To evaluate the performance of these algorithms in terms of

[1] https://epfl-lts2.github.io/gspbox-html/

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173

learning accuracy, we utilize the relative error (RelErr) and the F-score criteria. Let $\mathbf{L}_n^* \in \mathbb{R}^{p\times p}$ be the ground-truth Laplacian at frame n, and $\widehat{\mathbf{L}}_n \in \mathbb{R}^{p\times p}$ be the estimated one. We scale both matrices so that $\mathrm{tr}(\widehat{\mathbf{L}}_n) = \mathrm{tr}(\mathbf{L}_n^*) = p$. We also apply a threshold to the estimated Laplacian to nullify the smallest 1% of the weights. The relative error and the F-score measures are then defined as follows:

$$\text{RelErr} = \frac{\|\mathbf{L}_n^* - \widehat{\mathbf{L}}_n\|_F}{\|\mathbf{L}_n^*\|_F}, \qquad \text{F-score} = \frac{2\,\text{TP}}{2\,\text{TP} + \text{FP} + \text{FN}}.$$

In the above equations, TP, FP, and FN respectively denote the number of correctly identified connections in the original graph, the number of connections falsely identified in the estimated graph (not present in the original one), and the number of connections from the original graph that are missing in the estimated one.

1) Robust Graph Learning: We first evaluate our proposed method against the benchmark for time-varying graph learning using clean synthetic data. Visual representations of the time-varying graph weights across each time frame (for 400 edges) learned using different algorithms are shown in Fig. 3. In these figures, the horizontal axis represents the frame index n, and the vertical axis represents the index of the graph's edges. The top row shows the results for frame lengths of $T_n = 200$, while the bottom row shows the results for $T_n = 100$. As observed in Fig. 3, the time-varying weights of the graph learned using the proposed method align more closely with the ground-truth graph weights.

Next, we introduce noise and missing samples to the data to evaluate the robustness of the time-varying graph learning methods. Tables II and III present the single-frame performance of the algorithms averaged over all time frames (with fixed length $T_n = 200$) for varying data sampling rates and noise levels, respectively. The values in these tables report the mean $\pm$ standard deviation of the average frame performance for 20 random trials of this experiment for three performance metrics, including F-score (classification accuracy), relative error (reconstruction quality), and runtime in seconds (computational complexity). As shown in Tables II and III, the proposed method achieves superior performance in time-varying graph learning from corrupted data (containing noise and missing values). While not the most computationally efficient approach, its computational complexity remains comparable to several baseline methods. We will discuss this in more detail in part IV-A3.

2) Hyperparameter Selection: In this part, we examine the sensitivity of the proposed algorithm to the choice of the hyper-parameters. Figures 4 to 6, demonstrate the F-score performance of the proposed method in Alg. 1 versus different values of the hyper-parameters d, ρ, σ_e, γ and k. The error-bar plots in these figures are obtained by averaging the F-score performance results over 20 trials of random experiments with synthetically generated dataset (with $p = 100$ rows and $T = 1000$ columns). The proposed method, as shown in these figures, demonstrates lower sensitivity to the choice of γ compared to other hyper-parameters, which exhibit a more significant impact on performance. Based on these results, we choose the optimal hyper-parameter values that maximize the F-score as $\mathbf{d} = \mathbf{1}$, $\rho = 1$, $\sigma_\epsilon = 1$, $\nu = 3$, and $\gamma = 10$. The value of k, however, is determined by the number of clusters in the data. The synthetic data used in our experiment, is sampled from a stochastic block model with $K = 4$ clusters and hence, $k = 4$ is the optimal choice in this case.

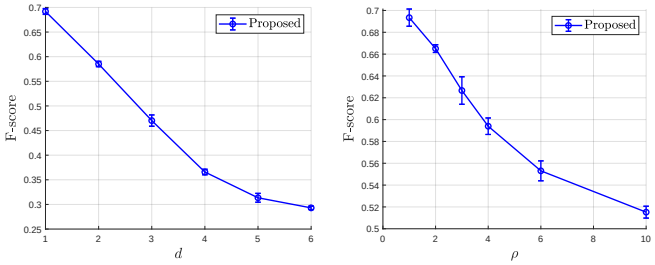


Fig. 4: Sensitivity of the proposed method to the parameters d (left) and ρ (right).

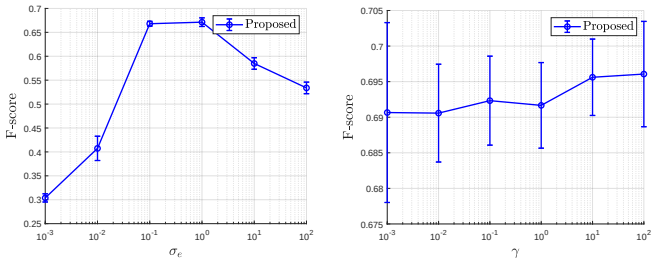


Fig. 5: Sensitivity of the proposed method to the parameters σ_e (left) and γ (right).

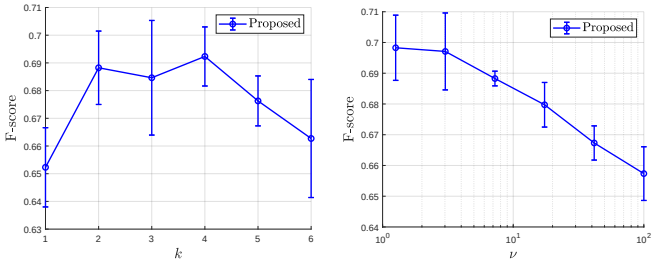


Fig. 6: Sensitivity of the proposed method to the parameters k (left) and ν (right).

3) Run-time Complexity: Fig. 7 depicts the run-time of our proposed method compared to the benchmark for different number of nodes. These results are obtained using MATLAB on a PC with an Intel® Core™ i7-12700K processor and 16 GB DDR4 RAM. Based on this figure and the computational complexity analysis given in Section III-C, the complexity of our proposed approach (which is $\mathcal{O}(p^3 + T_n p^2)$) more rapidly increases with the number of nodes compared to some baseline methods with a complexity of $\mathcal{O}(p^2)$ (e.g., Kalofolias et al.). To be able to learn k-component graphs, we directly deal with the eigenvalues and the eigenvectors of the Laplacian in each iteration, the computation of which is costly ($\mathcal{O}(p^3)$) and affects the complexity of our developed algorithm. Compared to the baseline, we also have an additional step corresponding to the reconstruction of the data matrix

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173

$\mathbf{X}$ in our algorithm (making it robust to missing samples and noise), which in turn adds to this complexity. Thus, the application of the proposed method can be limited for large-scale networks. However, for financial markets, e.g., SP500 index, the number of assets in the network is not very large (e.g., 500) and the proposed approach can still be applied to infer the clusters and how entities in each cluster interact with each other (as we will see in the next part). Additionally, for clean and complete data, the signal reconstruction step ($\mathbf{X}$-update step) is unnecessary and can be removed. This will lead to reduction in complexity as shown in Fig. 7 which compares the run-time of the proposed method with and without the $\mathbf{X}$-update step (the latter is labeled as 'Proposed (no $\mathbf{X}$-update)'). In conclusion, we obtain a more illustrative and more robust graph representation for the data compared to the traditional methods for time-varying learning with the cost of more computational complexity. By the way, we can always control the complexity of our method by adjusting the frame length parameter T_n (updating frequency). Nevertheless, our method is still faster than some baseline algorithms for online time-varying graph learning.

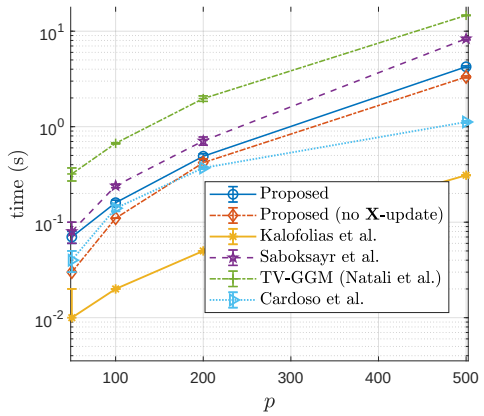


Fig. 7: Average single-frame run-time of the proposed method (for frame length $T_n = 200$) compared to the benchmark for different number of nodes, p.

B. Real Data

In this section, we utilize real-world data from financial markets, particularly the log-returns of the stocks in the S&P500 index. For this experiment, we select a subset of 100 stocks, categorized into $K = 8$ sectors (clusters), including "Utilities", "Real State", "Materials", "Industrials", "Health Care", "Financials", "Energy", and "Consumer Staples". The ground-truth labels of the sectors are determined by the GICS classification standard[2]. We subsequently compute the log-return of these stocks over a 1000-day period spanning from January 2016 to January 2020. Following this, we construct the matrix $\mathbf{X} \in \mathbb{R}^{p\times T}$ with $p = 100$ (number of stocks) and $T = 1000$ (number of days). We then partition data into non-overlapping frames of length $T_n = 200$, resulting in $N = 5$ data frames. Next, we evaluate the effectiveness of our method for two different applications: stock classification or clustering and portfolio design. For the clustering task, we employ several criteria to evaluate performance, including accuracy (ACC), purity [54], the modularity (MOD) [55], and the adjusted Rand index (ARI) [56]. Both accuracy and purity are calculated by determining the ratio of the true-positive (correctly classified) labels (TP) to p. However, there is a distinction between these two metrics: in accuracy, we consider the best ordering of the labels assigned to the inferred clusters (among all $K!$ permutations), whereas in purity, the label of each cluster is assumed to be the ground-truth label of the majority of the nodes in that cluster. The modularity is also a measure that evaluates how disjoint the nodes with different labels are (the higher the value, the more disjoint they are). The ARI is another metric used to evaluate the similarity between the true labels and the cluster labels. Here, the parameter ν in the proposed method is obtained by fitting a multivariate Student-t distribution to each data frame (using the `fitHeavyTail` R-package[3]).

1) Multi-component Graph Clustering: For clustering the data into $K = 8$ clusters, we first learn K-component graphs where each component represents one data cluster. To this end, we use our proposed method along with several existing benchmark algorithms for multi-component graph learning (which are static methods). These algorithms include the constrained Laplacian rank (CLR) method [52], the SGLA method[4] [19], the Fingraph algorithm[5] [48], and the method proposed by Javaheri et al. [53] for balanced clustering. Notably, the latter two methods are tailored for heavy-tailed data. For these benchmark algorithms, which are all designed for offline static graph learning, we provide the static graph learned from each data frame as the initial guess for graph learning in the next frame, i.e., $\mathbf{w}_{n+1}^0 = \hat{\mathbf{w}}_n$.

Fig. 8 illustrates the graphs learned from the last frame of data ($n = N = 5$). The colors of the nodes (stocks) indicate their respective ground-truth clusters (sector indices), while labels adjacent to certain nodes denote the misclassified stocks.

TABLE IV: Clustering performance of the graphs shown in Fig. 8. The values are reported for the last data frame (with $T_n = 200$). The time column displays the total runtime for each method.

	ACC ↑	Purity ↑	MOD ↑	ARI ↑	Time (s) ↓
CLR [52]	0.56	0.67	0.23	0.33	1.53
SGLA [19]	0.27	0.29	0.27	0.01	**1.21**
Fingraph [48]	0.49	0.53	0.47	0.21	2.38
Javaheri et al. [53]	0.61	0.69	0.38	0.32	2.70
Proposed ($k = 8$)	**0.69**	**0.78**	**0.62**	**0.58**	2.14

TABLE V: Clustering performance of the proposed method on sample S&P500 dataset for different choices of the frame length T_n. (the values are reported for the last data frame).

	ACC ↑	Purity ↑	MOD ↑	ARI ↑	Delay (days) ↓
$T_n = 100$	0.68	0.72	0.59	0.53	**100**
$T_n = 200$	0.69	0.78	**0.62**	0.58	200
$T_n = 500$	0.71	0.84	0.60	0.67	500
$T_n = 1000$	**0.79**	**0.89**	0.59	**0.75**	1000

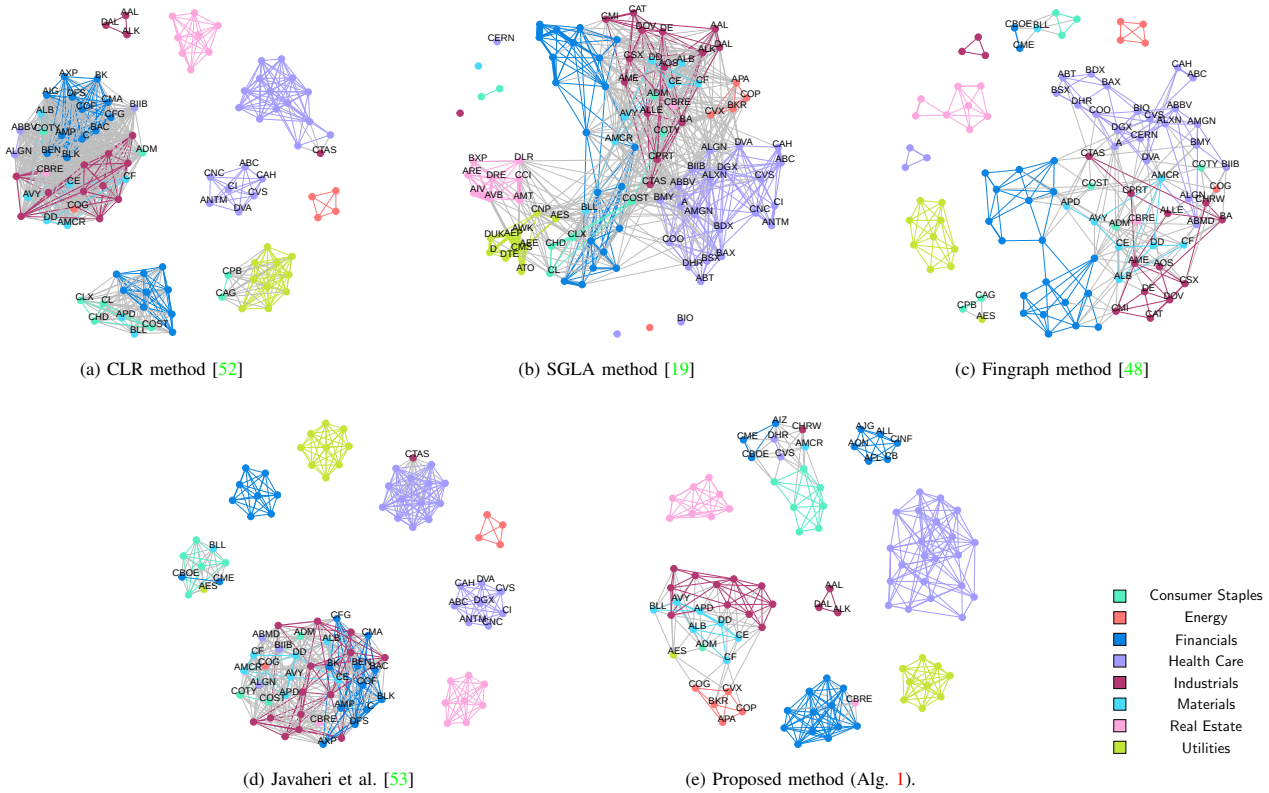
[2] https://www.msci.com/our-solutions/indexes/gics

[3] https://CRAN.R-project.org/package=fitHeavyTail

[4] https://CRAN.R-project.org/package=spectralGraphTopology

[5] https://CRAN.R-project.org/package=fingraph

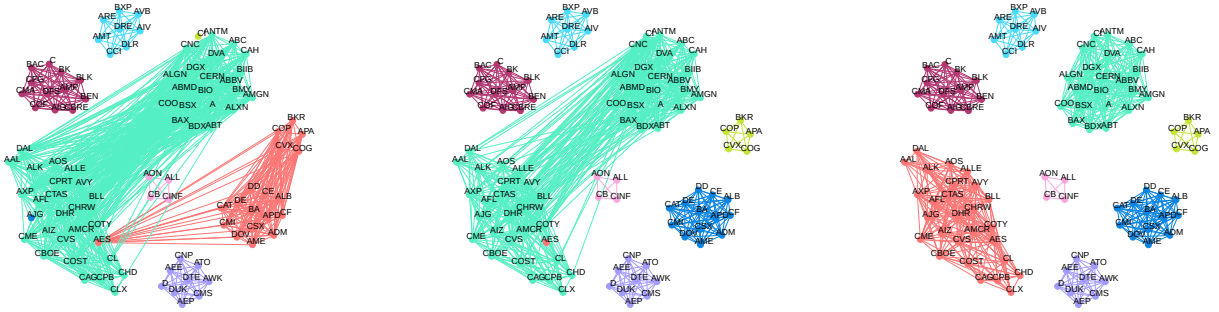
This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173



(a) CLR method [52] (b) SGLA method [19] (c) Fingraph method [48]

(d) Javaheri et al. [53] (e) Proposed method (Alg. 1).

Fig. 8: The K-component graphs learned from a sample experiment with financial data corresponding to the log-returns of 100 stocks in the S&P 500 index (comprising $K = 8$ sectors). The graphs are shown for the last data frame (with length $T_n = 200$).



(a) Graph learned from the 1st frame ($T_n = 200$). (b) Graph learned from the 2nd frame ($T_n = 200$). (c) Graph learned from the 3rd frame ($T_n = 200$).

Fig. 9: The time evolution of the graphs learned from a sample experiment with S&P500 data via the proposed method for the frame length of $T_n = 200$. Colors represent the inferred clusters ($K = 8$).

The clustering performance corresponding to these methods as well as the runtime of the algorithms is summarized in Table IV. The results indicate that the time-varying graph learned using the proposed algorithm (Alg. 1) exhibits better clustering performance by taking advantage of the graph temporal variations, compared to the static methods for graph learning. As expected, the performance of the proposed time-varying clustering method is improved through time-evolution as shown in Fig. 9. This is also visually depicted in Fig. 10, where the clustering performance is plotted against the frame number, illustrating its improvement over time. We then evaluate the performance of our proposed method for different choices of the frame length, namely $T_n = 100$, $T_n = 200$, $T_n = 500$, and $T_n = 1000$. The results are given in Table V, where the last column represents the number of days we have

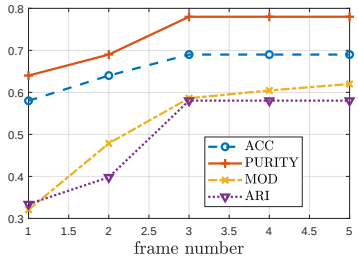


Fig. 10: Clustering performance of the graphs learned from a sample S&P500 dataset using our proposed method versus the frame number. The first few graphs are shown in Fig. 9.

to wait to update the graph learned form financial data. As illustrated in the table, the accuracy of the clustering improves as the frame length increases, due to the availability of more data samples and more accurate statistical estimates. However, this also requires access to more data and introduces larger delay. Therefore, depending on the application, a trade-off between delay and accuracy should be considered.

2) Spectral Graph Clustering: We then consider graph-based clustering based on unconstrained graph structures (with no Laplacian rank constraints). For this we use spectral graph clustering in combination with time-varying graph learning methods to cluster the stock data. We compare our method with benchmark algorithms for time-varying graph learning, including the methods proposed by Natali et al. (TV-GGM, TV-SBM) [34], [35] for online time-varying graph topology identification under the Gaussian graphical model, the time-varying graph learning method proposed by Cardoso et al. [28], the online time-varying graph topology inference method by Saboksayr et al. [41], and the time-varying graph learning algorithm by Kalofolias et al. [25]. These methods are not applicable for learning K-component graphs (based on which we can cluster the data). Therefore, we utilize the spectral clustering approach [57] for node classification. Specifically, we apply K-means clustering [58] to the rows of the matrix formed by the eigenvectors of the inferred Laplacian corresponding to the K smallest eigenvalues. Table VI presents the performance comparison of the proposed method with benchmark time-varying graph learning algorithms for clustering financial (S&P 500) data. Here, the frame length is fixed at $T_n = 200$ for all methods, and the reported values represent the clustering performance on the last frame of the data ($n = N$). This table presents the results of our proposed method for the connected graph case with $k = 1$ (without Laplacian rank constraint). A comparison of Tables IV and VI reveals that our proposed method's multi-component graph structure version (with $k = 8$) achieves superior clustering performance. This improvement highlights the significance of rank (spectral) constraint in enhancing the clustering performance. In both clustering approaches, the proposed method is also shown to outperform the baseline, attributed to its capability to learn graphs from heavy-tailed data.

3) Multiple Random Experiments: To have a more comprehensive evaluation of our proposed method's performance, we conduct multiple randomized simulations across diverse datasets and report averaged results. Specifically, we select 20 random datasets of S&P 500 log-returns (from 2014-01-01 to 2024-01-01), each containing $p = 100$ randomly chosen stocks over $T = 1000$ day periods, partitioned into non-overlapping frames. We evaluate mean clustering performance against benchmarks across all frames. Figures 11–14 present the average single-frame performance measures of clustering with error bars showing mean values (central markers) and standard deviations (vertical bars). These results conclusively demonstrate the superiority of the proposed method for dynamic heavy-tailed data clustering compared to the baseline.

4) Portfolio Design: Next, we explore the application of our time-varying graph learning method for investment portfolio

TABLE VI: Clustering performance comparison between the proposed method and the benchmark algorithms for learning time-varying graphs, where K-means is employed for spectral clustering (the results are reported for last frame of a sample S&P500 dataset with frame length $T_n = 200$). The time column displays the total runtime for each method.

	ACC ↑	Purity ↑	MOD ↑	ARI ↑	Time (s) ↓
Proposed ($k = 1$)	**0.63**	**0.73**	0.58	**0.52**	2.02
Kalofolias et al. [25]	0.57	0.67	**0.59**	0.16	**0.18**
Cardoso et. al [28]	0.47	0.49	0.36	0.41	1.45
TV-GGM [34]	0.29	0.31	0	0.08	6.76
TV-SBM [35]	0.61	0.63	0.58	0.36	522.58
Saboksayr et al. [41]	0.41	0.44	0.44	0.12	2.26

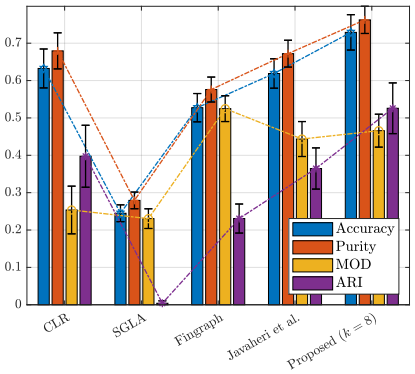


Fig. 11: Average single-frame performance of the proposed method for different datasets compared to the baseline K-component graph learning methods (for $T_n = 200$ and $K = 8$).

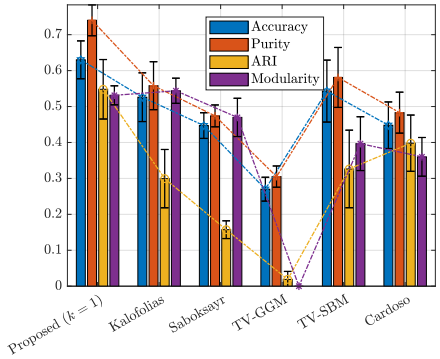


Fig. 12: Average single-frame clustering performance of the proposed method (with $k = 1$) compared to the baseline time-varying graph learning methods for different datasets. Here, the frame length is $T_n = 200$ and we apply spectral graph clustering on the learned graphs.

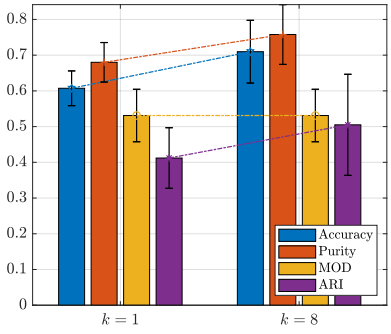


Fig. 13: Average single-frame clustering performance of the proposed method with $k = 1$ (via spectral graph clustering) and $k = 8$ (via multi-component graph clustering). Here the frame length is $T_n = 200$.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3588173

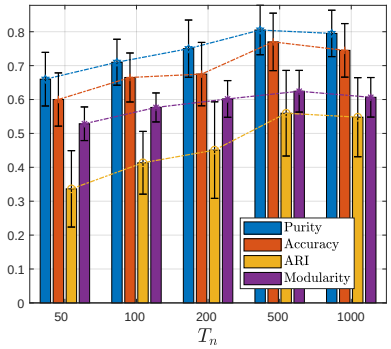


Fig. 14: Average single-frame clustering performance of the proposed method (with $k = 1$) versus the frame length. for different datasets. Here, we have used spectral graph clustering on the learned graph.

design. We consider a dynamic portfolio design strategy based on maximizing the ratio of the portfolio return over the portfolio graph smoothness. Let $\mathbf{L}_n$ denote the time-varying graph learned by the proposed method (Algorithm 1) given the n-th data frame. Let $\mathbf{u}_n$ denote the portfolio weights at time frame n and assume the expected value (mean) and the covariance matrix of the stock returns for the n-th data frame are given as $\hat{\boldsymbol{\mu}}_n$ and $\hat{\boldsymbol{\Sigma}}_n$. Here, we use Student-$t$ robust estimators for $\hat{\boldsymbol{\mu}}_n$ and $\hat{\boldsymbol{\Sigma}}_n$ (using the `fitHeavyTail` R-package similar to the previous part). The Maximum Sharpe Ratio Portfolio (MSRP) design scheme aims at maximizing the Sharpe ratio (S) [59] defined as the ratio of the portfolio expected return over the volatility, i.e., $\mathrm{S} = \frac{\hat{\boldsymbol{\mu}}_n^\top \mathbf{u}_n}{\mathbf{u}_n^\top \hat{\boldsymbol{\Sigma}}_n \mathbf{u}_n}$. Using the Schaible transform [60], the dynamic MSRP optimization problem can be reformulated as:

$$\mathbf{u}^\star_{\mathrm{MSRP},n} = \underset{\mathbf{u}_n \geq \mathbf{0},\, \hat{\boldsymbol{\mu}}_n^\top \mathbf{u}_n = 1}{\operatorname{argmin}} \; \mathbf{u}_n^\top \hat{\boldsymbol{\Sigma}}_n \mathbf{u}_n \tag{32}$$

where $\mathbf{u}^\star_{\mathrm{MSRP},n}$ denotes the optimal MSRP weights at time frame n. Inspired by the notion of the Sharpe ratio, we propose the Graph-based Ratio (GR) as

$$\mathrm{GR} \triangleq \frac{\hat{\boldsymbol{\mu}}_n^\top \mathbf{u}_n}{\sqrt{\mathbf{u}_n^\top \mathbf{L}_n \mathbf{u}_n}}.$$

To maximize this ratio, we design a portfolio referred to as the Maximum Time-Varying Graph Ratio Portfolio (MTVGRP). The problem for this portfolio design is stated as follows:

$$\mathbf{u}^\star_{\mathrm{MTVGRP},n} = \underset{\mathbf{u}_n \geq \mathbf{0},\, \hat{\boldsymbol{\mu}}_n^\top \mathbf{u}_n = 1}{\operatorname{argmin}} \; \mathbf{u}_n^\top \mathbf{L}_n \mathbf{u}_n \tag{33}$$

Consider the returns of 50 randomly chosen stocks from the S&P500 index over the period 2010-12-01 to 2018-12-01. From this dataset, we select 100 different subsets (time intervals), each of length $T = 504$ days, with different starting time indices. Each dataset is partitioned into frames of length $T_n = 200$ days with 180 days overlap. We then design dynamic portfolios (with re-optimization frequency of 20 days) based on our proposed MTVGRP scheme, and evaluate the performance using the portfolio backtest[6] package in R. We compare our design scheme with the dynamic MSRP portfolio and the Equally Weighted Portfolio (EWP). The weights for the MSRP are obtained by solving (32) and the EWP weights are given as $\mathbf{u}^\star_{\mathrm{EWP},n} = 1/p$. Table VII shows the results of this backtest in terms of different performance criteria, including the Sharpe ratio, the annualized return, the annualized volatility, and the maximum drawdown. From this table, it is implied that the proposed method delivers better performance through the (dynamic) time-varying graph-based portfolio design scheme. This is also evident in Fig. 15 where the Sharpe ratio and the maximum drawdown of the portfolios in Table VII are compared with the market index.

[6] https://CRAN.R-project.org/package=portfolioBacktest

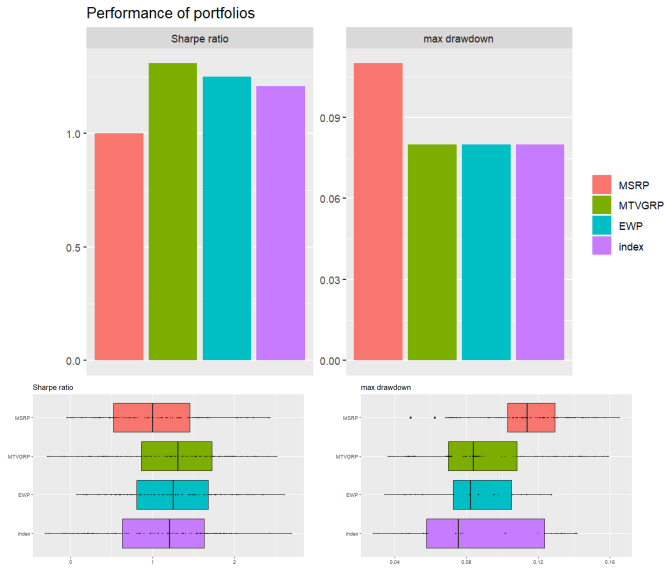


Fig. 15: Sharpe ratio (left) and the maximum draw-down (right) performance of the proposed MTVGRP portfolio compared to the market index, the MSRP, and the EWP portfolios. The barplots on the top represent the mean values of Sharpe ratio (left) and maximum drawdown (right) and the boxplots on the bottom depict the distribution of these performance measures.

TABLE VII: Average backtesting results of the proposed MTVGRP portfolio compared to the MSRP and the EWP portfolios.

	Ann. Return ↑	Ann. Volatility ↓	Sharpe Ratio ↑	Max Drawdown ↓
MTVGRP (Proposed)	**0.14**	**0.11**	**1.31**	**0.08**
MSRP	0.13	0.14	1.00	0.11
EWP (Uniform)	0.13	0.11	1.25	0.08

V. Conclusion

This paper explores the problem of learning time-varying graphs specifically designed for heavy-tailed data. We propose a novel approach for time-varying graph learning that is tailored to infer graph-structured models capable of effectively capturing heavy-tailed distributions. Our proposed approach employs a probabilistic model to formulate the problem of learning time-varying graphs. We also incorporate spectral constraints into the problem, enabling us to learn multi-component graphs suitable for clustering. We present a solution based on a maximum-a-posteriori estimation framework using a semi-online strategy, wherein a single data frame is utilized to update the graph. To demonstrate the effectiveness and robustness of our method in graphical modeling of time-

varying heavy-tailed data, particularly within financial markets, we conduct numerical experiments using both synthetic and real datasets.

Appendix A
Proof of Proposition 1

Proof. The MAP estimation rule can be expressed as follows:

$$\begin{aligned}\min_{\mathbf{w}_n\geq\mathbf{0},\,\mathbf{a}\geq\mathbf{0},\,\mathbf{X}_n} \quad & -\log p(\mathbf{w}_n,\mathbf{a},\mathbf{X}_n|\mathbf{w}_{n-1},\mathbf{Y}_n,\mathbf{M}_n) = \\ & -\log p(\mathbf{Y}_n|\mathbf{X}_n,\mathbf{M}_n) - \log p(\mathbf{X}_n|\mathbf{w}_n) \\ & -\log p(\mathbf{w}_n|\mathbf{w}_{n-1},\mathbf{a}) - \log p(\mathbf{a}) + \text{const} \\ \text{s.t.} \quad & \mathbf{w}_n \in \Omega_{\mathbf{w}}.\end{aligned}$$

Now, given the Gaussian distribution of the measurement noise, we may write:

$$\begin{aligned}p(\mathbf{Y}_n|\mathbf{X}_n,\mathbf{M}_n) &= \prod_{t\in F_n} p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{m}_t) \\ &= C_0 \exp\left(-\frac{1}{2\sigma_n^2}\left\|\mathbf{Y}_n - \mathbf{M}_n\odot\mathbf{X}_n\right\|_F^2\right),\end{aligned}$$

where C_0 is a constant. Moreover, assuming the Student-t distribution in (10) for the data given the underlying graph model, we have:

$$\begin{aligned}p(\mathbf{X}|\mathbf{w}_n) &= \prod_{t\in F_n} p(\mathbf{x}_t|\mathbf{w}_n) \\ &= C_1 \prod_{t\in F_n} \det{}^*(\mathcal{L}\mathbf{w}_n)\left(1+\frac{\mathbf{x}_t^\top\mathcal{L}\mathbf{w}_n\mathbf{x}_t}{\nu}\right)^{-(\nu+p)/2}.\end{aligned}$$

with C_1 being another constant.

Now, let $\mathbf{v}_n = \mathbf{a}\odot\mathbf{w}_{n-1} \geq \mathbf{0}$. Then, we may write:

$$p(\mathbf{w}_n|\mathbf{w}_{n-1},\mathbf{a}) = p(\mathbf{w}_n|\mathbf{v}_n) = \prod_i p(w_n(i)|v_n(i)).$$

Using the non-negative VAR equation $w_n(i) = (v_n(i) + \epsilon_n(i))_+$, we have:

$$p(w_n(i)|v_n(i)) = \begin{cases} \mathbb{P}[\epsilon_n(i) = w_n(i) - v_n(i)] & w_n(i) > 0 \\ \mathbb{P}[\epsilon_n(i) \leq -v_n(i)] & w_n(i) = 0 \end{cases}$$

Assuming i.i.d. Laplace distribution for the elements of $\boldsymbol{\epsilon}_n$ as in (9), we get:

$$\begin{aligned}p(w_n(i)|v_n(i)) &= \begin{cases} \frac{1}{2\sigma_\epsilon}\exp\left(-\frac{|w_n(i)-v_n(i)|}{\sigma_\epsilon}\right) & w_n(i) > 0 \\ \frac{1}{2}\exp\left(\frac{-v_n(i)}{\sigma_\epsilon}\right) & w_n(i) = 0 \end{cases} \\ &= \frac{1}{2\sigma_\epsilon^{\mathbb{1}(w_n(i)>0)}}\exp\left(-\frac{|w_n(i)-v_n(i)|}{\sigma_\epsilon}\right).\end{aligned}$$

Hence, we obtain:

$$p(\mathbf{w}_n|\mathbf{w}_{n-1},\mathbf{a}) = \frac{1}{2^p}\frac{1}{\sigma_\epsilon^{\|\mathbf{w}_n\|_0}}\exp\left(-\frac{1}{\sigma_\epsilon}\left\|\mathbf{w}_n - \mathbf{a}\odot\mathbf{w}_{n-1}\right\|_1\right).$$

Finally, plugging these probabilities in (12), also considering exponential distribution for the VAR parameters $\mathbf{a}$ as in (9), we can obtain (13) after simplification. □

Appendix B
Proof of Proposition 2

Proof. Let $g(\mathbf{x}_t) = \log\left(1+\frac{\mathbf{x}_t^\top\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t}{\nu}\right)$. Then, using the inequality $\log(x) \leq x-1,\ \forall x>0$, we have:

$$\begin{aligned}g(\mathbf{x}_t) &= \log\left(1+\frac{\mathbf{x}_t^\top\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t}{\nu}\right) \\ &\leq \log\left(1+\frac{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l}{\nu}\right) + \frac{\mathbf{x}_t^\top\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t+\nu}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu} - 1 \\ &= g(\mathbf{x}_t^l) - 1 + \mathbf{x}_t^\top\frac{\mathcal{L}\mathbf{w}_n^{l+1}}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu}\mathbf{x}_t + \frac{\nu}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu} \\ &= \frac{1}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu}\mathbf{x}_t^\top\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t + h(\mathbf{x}_t^l),\end{aligned}$$

where $h(\mathbf{x}_t^l)$ is a constant. Then, we may write:

$$\begin{aligned}f_{\mathbf{x}_t}(\mathbf{x}_t) &= \frac{1}{T_n\sigma_n^2}\left\|\mathrm{Diag}(\mathbf{m}_t)\mathbf{x}_t - \mathbf{y}_t\right\|^2 + \frac{p+\nu}{T_n}g(\mathbf{x}_t) \\ &\leq \frac{1}{T_n}\left(\mathbf{x}_t^\top\mathbf{Q}_t\mathbf{x}_t - 2\mathbf{c}_t^\top\mathbf{x}_t\right) + r(\mathbf{x}_t^l),\end{aligned}$$

with $\mathbf{Q}_t$ and $\mathbf{c}_t$ given in (26) and $r(\mathbf{x}_t^l) = \frac{p+\nu}{T_n}h(\mathbf{x}_t^l) + \frac{\|\mathbf{y}_t\|^2}{T_n\sigma_n^2}$. Now, for $\tau > \lambda_{\max}(\mathbf{Q}_t)$, we may propose another majorization function as follows:

$$\begin{aligned}f_{\mathbf{x}}(\mathbf{x}_t) &\leq \frac{1}{T_n}\Big(\mathbf{x}_t^\top\mathbf{Q}_t\mathbf{x}_t + (\mathbf{x}_t-\mathbf{x}_t^l)^\top(\tau\mathbf{I}-\mathbf{Q}_t)(\mathbf{x}_t-\mathbf{x}_t^l) \\ &\qquad - 2\mathbf{c}_t^\top\mathbf{x}_t\Big) + r(\mathbf{x}_t^l) \\ &= \frac{\tau}{T_n}\left\|\mathbf{x}_t - \mathbf{x}_t^l + \frac{\mathbf{Q}_t\mathbf{x}_t^l - \mathbf{c}_t}{\tau}\right\|^2 + C(\mathbf{x}_t^l),\end{aligned}$$

where $C(\mathbf{x}_t^l)$ is a constant. To find $\tau > \lambda_{\max}(\mathbf{Q}_t)$, we write:

$$\begin{aligned}\lambda_{\max}(\mathbf{Q}_t) &= \lambda_{\max}\left(\tfrac{1}{\sigma_n^2}\mathrm{Diag}(\mathbf{m}_t) + \tfrac{p+\nu}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu}\mathcal{L}\mathbf{w}_n^{l+1}\right) \\ &\leq \tfrac{1}{\sigma_n^2}\lambda_{\max}\left(\mathrm{Diag}(\mathbf{m}_t)\right) + \tfrac{(p+\nu)\lambda_{\max}\left(\mathcal{L}\mathbf{w}_n^{l+1}\right)}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu} \\ &= \tfrac{1}{\sigma_n^2} + \tfrac{p+\nu}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu}\lambda_{\max}\left(\mathcal{L}\mathbf{w}_n^{l+1}\right),\end{aligned}$$

where we applied Weyl's inequality [61] in the first expression. Hence, it suffices to choose $\tau \geq \frac{1}{\sigma_n^2} + \frac{p+\nu}{\mathbf{x}_t^{l\top}\mathcal{L}\mathbf{w}_n^{l+1}\mathbf{x}_t^l+\nu}\lambda_{\max}\left(\mathcal{L}\mathbf{w}_n^{l+1}\right)$. □

References

[1] A. Javaheri and D. P. Palomar, "Learning time-varying graphs for heavy-tailed data clustering," in *2024 32nd European Signal Processing Conference (EUSIPCO)*, 2024, pp. 2472–2476.

[2] W. Campbell, C. Dagli, and C. Weinstein, "Social Network Analysis with Content and Graphs," *Lincoln Laboratory Journal*, vol. 20, no. 1, pp. 62–81, 2013.

[3] X. Zhang, Y. Sun, H. Liu, Z. Hou, F. Zhao, and C. Zhang, "Improved clustering algorithms for image segmentation based on non-local information and back projection," *Information Sciences*, vol. 550, pp. 129–144, Mar. 2021.

[4] H. Tamura, K. Nakano, M. Sengoku, and S. Shinoda, "On Applications of Graph/Network Theory to Problems in Communication Systems," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 5, no. 1, pp. 15–21, Jan. 1970.

[5] J. V. de M. Cardoso and D. P. Palomar, "Learning undirected graphs in financial markets," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 741–745.

[6] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

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

[8] H. Rue and L. Held, *Gaussian Markov random fields: theory and applications*, Number 104 in Monographs on statistics and applied probability. Chapman & Hall/CRC, Boca Raton, 2005.

[9] J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Nonconvex sparse graph learning under Laplacian constrained graphical model," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 7101–7113.

[10] J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Minimax estimation of Laplacian constrained precision matrices," in *International Conference on Artificial Intelligence and Statistics*, 2021, vol. 130, pp. 3736–3744.

[11] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph Learning From Data Under Laplacian and Structural Constraints," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825–841, Sept. 2017.

[12] C. Zhang, D. Florencio, and P. A. Chou, "Graph signal processing - a probabilistic framework," Tech. Rep. MSR-TR-2015-31, April 2015.

[13] J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Does the ℓ_1-norm learn a sparse graph under Laplacian constrained graphical models?," *arXiv preprint arXiv:2006.14925*, 2020.

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

[15] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graphs," in *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society*, Portland, Oregon, United States, Aug. 2010, pp. 778–784.

[16] L. Zhao, Y. Wang, S. Kumar, and D. P. Palomar, "Optimization Algorithms for Graph Laplacian Estimation via ADMM and MM," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4231–4244, Aug. 2019.

[17] J. V. de M. Cardoso, J. Ying, and D. P. Palomar, "Learning bipartite graphs: Heavy tails and multiple components," in *Advances in Neural Information Processing Systems*, 2022, vol. 35, pp. 14044–14057.

[18] A. Javaheri and D. P. Palomar, "Clustering of Incomplete Data via a Bipartite Graph Structure," May 2025, arXiv:2505.08594 [cs].

[19] S. Kumar, J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "A Unified Framework for Structured Graph Learning via Spectral Constraints," *Journal of Machine Learning Research*, vol. 21, no. 22, pp. 1–60, 2020.

[20] D. Kaplan, *Structural Equation Modeling (2nd ed.): Foundations and Extensions*, SAGE Publications, Inc., 2455 Teller Road, Thousand Oaks California 91320 United States, 2009.

[21] J. Songsiri and L. Vandenberghe, "Topology Selection in Graphical Models of Autoregressive Processes," *Journal of Machine Learning Research*, vol. 11, no. 91, pp. 2671–2705, 2010.

[22] A. Bolstad, B. D. Van Veen, and R. Nowak, "Causal Network Inference Via Group Sparse Regularization," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2628–2641, June 2011.

[23] J. Mei and J. M. F. Moura, "Signal Processing on Graphs: Causal Modeling of Unstructured Data," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017.

[24] A. Javaheri, A. Amini, F. Marvasti, and D. P. Palomar, "Learning Spatiotemporal Graphical Models From Incomplete Observations," *IEEE Transactions on Signal Processing*, vol. 72, pp. 1361–1374, 2024.

[25] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, Mar. 2017, pp. 2826–2830.

[26] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, "Network inference via the time-varying graphical lasso," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2017, p. 205–213.

[27] K. Yamada, Y. Tanaka, and A. Ortega, "Time-Varying Graph Learning with Constraints on Graph Temporal Variation," Jan. 2020, arXiv:2001.03346.

[28] J. V. de M. Cardoso and D. P. Palomar, "Learning Undirected Graphs in Financial Markets," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, Nov. 2020, pp. 741–745.

[29] A. Natali, E. Isufi, M. Coutino, and G. Leus, "Online Graph Learning From Time-Varying Structural Equation Models," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, Oct. 2021, pp. 1579–1585, ISSN: 2576-2303.

[30] R. Hamon, P. Borgnat, P. Flandrin, and C. Robardet, "Tracking of a dynamic graph using a signal theory approach : application to the study of a bike sharing system," Sept. 2013, p. 101.

[31] F. Harary and G. Gupta, "Dynamic graph models," *Mathematical and Computer Modelling*, vol. 25, no. 7, pp. 79–87, Apr. 1997.

[32] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying Graph Learning Based on Sparseness of Temporal Variation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5411–5415.

[33] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian Matrix in Smooth Graph Signal Representations," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.

[34] A. Natali, M. Coutino, E. Isufi, and G. Leus, "Online Time-Varying Topology Identification Via Prediction-Correction Algorithms," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 2021, pp. 5400–5404, ISSN: 2379-190X.

[35] A. Natali, E. Isufi, M. Coutino, and G. Leus, "Learning Time-Varying Graphs From Online Data," *IEEE Open Journal of Signal Processing*, vol. 3, pp. 212–228, 2022.

[36] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A Class of Prediction-Correction Methods for Time-Varying Convex Optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4576–4591, Sept. 2016.

[37] B. Baingana and G. B. Giannakis, "Tracking Switched Dynamic Network Topologies From Information Cascades," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 985–997, Feb. 2017.

[38] S. Vlaski, H. P. Maretić, R. Nassif, P. Frossard, and A. H. Sayed, "Online Graph Learning from Sequential Data," in *2018 IEEE Data Science Workshop (DSW)*, June 2018, pp. 190–194.

[39] R. Shafipour and G. Mateos, "Online Topology Inference from Streaming Stationary Graph Signals with Partial Connectivity Information," *Algorithms*, vol. 13, no. 9, pp. 228, Sept. 2020.

[40] R. Money, J. Krishnan, and B. Beferull-Lozano, "Online Non-linear Topology Identification from Graph-connected Time Series," in *2021 IEEE Data Science and Learning Workshop (DSLW)*, June 2021, pp. 1–6.

[41] S. S. Saboksayr, G. Mateos, and M. Cetin, "Online Graph Learning under Smoothness Priors," *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 1820–1824, Aug. 2021.

[42] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "Online learning of time-varying signals and graphs," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5230–5234.

[43] A. Buciulea, M. Navarro, S. Rey, S. Segarra, and A. G. Marques, "Online Network Inference from Graph-Stationary Signals with Hidden Nodes," Sept. 2024, arXiv:2409.08760.

[44] X. Zhang and Q. Wang, "Graph learning from incomplete graph signals: From batch to online methods," *Signal Processing*, vol. 226, pp. 109663, 2025.

[45] S. I. Resnick, *Heavy-tail phenomena: probabilistic and statistical modeling*, Springer series in operations research and financial engineering. Springer, New York, NY [Heidelberg], 2007.

[46] J. Paratte, "Graph-based Methods for Visualization and Clustering," 2017, Publisher: Lausanne, EPFL.

[47] A. Javaheri, A. Amini, F. Marvasti, and D. P. Palomar, "Joint Signal Recovery and Graph Learning from Incomplete Time-Series," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2024, pp. 13511–13515.

[48] J. V. de M. Cardoso, J. Ying, and D. P. Palomar, "Graphical Models in Heavy-Tailed Markets," in *Advances in Neural Information Processing Systems*, 2021, vol. 34, pp. 19989–20001.

[49] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[50] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[51] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[52] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The Constrained Laplacian Rank algorithm for graph-based clustering," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, Feb. 2016, AAAI'16, pp. 1969–1976.

[53] A. Javaheri, J. V. de M. Cardoso, and D. P. Palomar, "Graph Learning for Balanced Clustering of Heavy-Tailed Data," in *2023 IEEE 9th International Workshop on Computational Advances in Multi-Sensor*

Adaptive Processing (CAMSAP), Herradura, Costa Rica, Dec. 2023, pp. 481–485.

[54] B. Everitt, Ed., *Cluster analysis*, Wiley series in probability and statistics. Wiley, Chichester, West Sussex, U.K, 5th ed edition, 2011.

[55] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, June 2006.

[56] W. M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, Dec. 1971.

[57] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2001, NIPS'01, p. 849–856.

[58] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, 1967, pp. 481–485.

[59] W. F. Sharpe, "Mutual Fund Performance," *The Journal of Business*, vol. 39, no. 1, pp. 119–138, 1966.

[60] S. Schaible, "Parameter-free convex equivalent and dual programs of fractional programming problems," *Zeitschrift für Operations Research*, vol. 18, no. 5, pp. 187–196, Oct. 1974.

[61] A. J. Laub, *Matrix analysis for scientists and engineers*, Society for Industrial and Applied Mathematics, Philadelphia, 2005.

Amirhossein Javaheri (Graduate Student Member, IEEE) received his B.Sc. and M.Sc. degrees in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2014 and 2016, respectively. He is pursuing a dual PhD program in Electronic and Computer Engineering, jointly offered by Sharif University of Technology and the Hong Kong University of Science and Technology. His research focuses on graphs, signal processing, optimization, machine learning, statistics, and data analytics. He has received the Outstanding Bachelor's Thesis Award from Sharif University of Technology, the Iranian National Elites Foundation Award, the Hong Kong University of Science and Technology PhD scholarship and the Student Travel Grant from European Association for Signal Processing.

Jiaxi Ying (Member, IEEE) received his Ph.D. degree from the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology, Hong Kong, in 2022. He is currently a postdoctoral fellow at the Department of Mathematics, Hong Kong University of Science and Technology. He was a recipient of the Hong Kong Research Grants Council Postdoctoral Fellowship, the Outstanding Master's Thesis Award of Chinese Institute of Electronics, the Excellent Master Thesis in Fujian Provinc, the HKUST RedBird PhD Scholarship Program, and the NeurIPS Scholar Award. His research interests are mainly on the intersection of optimization, machine learning, signal processing, and statistics.

Daniel P. Palomar Daniel P. Palomar (Fellow, IEEE) received the Electrical Engineering and Ph.D. degrees from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1998 and 2003, respectively, and was a Fulbright Scholar at Princeton University during 2004-2006. He is a Professor in the Department of Electronic & Computer Engineering and in the Department of Industrial Engineering & Decision Analytics at the Hong Kong University of Science and Technology (HKUST), Hong Kong, which he joined in 2006. He had previously held several research appointments, namely, at King's College London (KCL), London, UK; Stanford University, Stanford, CA; Telecommunications Technological Center of Catalonia (CTTC), Barcelona, Spain; Royal Institute of Technology (KTH), Stockholm, Sweden; University of Rome "La Sapienza", Rome, Italy; and Princeton University, Princeton, NJ. His current research interests include data analytics, optimization methods, and deep learning in financial systems. Dr. Palomar is a EURASIP Fellow, an IEEE Fellow, a recipient of a 2004/06 Fulbright Research Fellowship, the 2004, 2015, and 2020 (co-author) Young Author Best Paper Awards by the IEEE Signal Processing Society, the 2015-16 HKUST Excellence Research Award, the 2002/03 best Ph.D. prize in Information Technologies and Communications by the Technical University of Catalonia (UPC), the 2002/03 Rosina Ribalta first prize for the Best Doctoral Thesis in Information Technologies and Communications by the Epson Foundation, and the 2004 prize for the best Doctoral Thesis in Advanced Mobile Communications by the Vodafone Foundation and COIT. He has been a Guest Editor of the IEEE Journal of Selected Topics in Signal Processing 2016 Special Issue on "Financial Signal Processing and Machine Learning for Electronic Trading", an Associate Editor of IEEE Transactions on Information Theory and of IEEE Transactions on Signal Processing, a Guest Editor of the IEEE Signal Processing Magazine 2010 Special Issue on "Convex Optimization for Signal Processing," the IEEE Journal on Selected Areas in Communications 2008 Special Issue on "Game Theory in Communication Systems," and the IEEE Journal on Selected Areas in Communications 2007 Special Issue on "Optimization of MIMO Transceivers for Realistic Communication Networks."

Farokh Marvasti (Life Senior Member, IEEE) received his B.Sc., M.Sc., and PhD degrees all from Rensselaer Polytechnic Institute (USA) in 1970, 1971 and 1973, respectively. He has worked, consulted and taught in Bell Labs, University of California Davis, Illinois Institute of Technology, University of London, and Kings College London. He was one of the editors and associate editors of the IEEE Transactions on Communications and Signal Processing from 1990-1997. He has published 2 books and 5 book chapters, about 200 Journal papers and several hundred conference papers. Dr Marvasti was a professor at Sharif University of Technology and the director of Advanced Communications Research Institute (ACRI) before he retired in 2021. He spent his sabbatical leave at the Communications and Information Systems Group of University College London (UCL) in 2013. Prof. Marvasti received a distinguished award from the Iranian Academy of Sciences in 2014 and a 5-year term chair position from Iranian National Science foundation in 2015. He was also appointed as a distinguished researcher by IEEE Iran Chapter in 2018. Prof. Marvasti has organised two concerts in London, East Meets West in 2003 and Gathering of the Birds in 2023.