Multiview Graph Learning Based on Node Perturbation Model

Mohammad Al-Wardat and Selin Aviyente
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, USA
e-mail: alwardat@msu.edu, aviyente@msu.edu

2024 58th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-5405-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/IEEECONF60004.2024.10942883

***Abstract*—Graph topology inference, i.e., learning graphs from a given set of nodal observations, is a significant task in many application domains. Existing approaches are mostly limited to learning a single graph assuming that the observed data is homogeneous. This is problematic because many modern datasets are heterogeneous or mixed and involve multiple related graphs, i.e., multiview graphs. Recent work proposing to learn multiview graphs ensures the similarity of learned view graphs through either pairwise regularization or consensus-based regularization. In most of the existing approaches, the similarities and differences between networks are assumed to be driven by individual edges. However, a node-based approach provides a more intuitive interpretation of the network differences. In this paper, we introduce a node perturbation model to learn multiview graph Laplacians with the assumption that the differences between the K networks are due to individual nodes that are perturbed across views. An optimization framework is formulated to simultaneously learn the graph Laplacians of K views with the assumption that the observed signals are smooth with respect to the underlying graph Laplacians. The proposed method is compared to existing edge-based multiview Laplacian learning methods as well as joint graphical Lasso based structure inference methods.**

***Index Terms*—Graph learning, perturbation model, smoothness, multiview graphs.**

I. Introduction

Graphs and graph-structured data are encountered in a variety of machine learning and signal processing applications. Some examples include social networks, gene regulatory networks and brain networks. While some of these applications such as social and traffic networks come with the graphical structure, in other cases such as in biological networks the graphical structure needs to be inferred from the observed signals. A variety of graph learning/inference methods have been proposed in both the signal processing and statistics literature [1]–[3].

Existing graph inference approaches are mostly limited to homogeneous datasets, where the observed graph signals are assumed to be identically distributed and defined on a single graph. In many applications, the data may be heterogeneous or mixed and come from multiple related graphs, i.e., multiview graphs. In these situations, learning the topology of each view by incorporating the relationships among views can improve the performance [4]–[6].

Existing methods for joint graphical structure inference are mostly based on Gaussian graphical models. These methods extend graphical lasso [3] to the multiview case by employing various penalty terms to exploit the common characteristics shared by the different views [5], [7]–[11]. One prominent example of this approach is the joint graphical lasso [5], where fused or group lasso penalties are used to encourage topological similarity across views. However, these methods are limited by the assumption that the observed graph signals are Gaussian which is usually not true for real-world applications. Furthermore, they learn the precision matrices without imposing any graph structure constraints on the learned views. Recently, these joint learning approaches have been extended to jointly learn multiple graph Laplacian matrices instead of precision matrices [12]–[14].

However, all of these methods quantify the pairwise similarity between the views based on edge similarity. In many settings, such as gene regulatory networks [9], the differences between views may be better explained through the changes in the connectivity of a few nodes. This way of modeling the differences imposes a structure and provides an intuitive interpretation of the network differences.

In this paper, we introduce a joint graph Laplacian learning framework where the differences across the views are assumed to be driven by the perturbation to the individual nodes' connectivity across views. Based on this assumption, we introduce a Laplacian learning framework using the smoothness criterion, i.e., the graph signals are smooth or low frequency with respect to the underlying graph structure, with a regularization term that captures the node-based similarity across views. We focus on learning graphs where each view is assumed to be a perturbed version of an underlying graph by changing the connectivity of r nodes with $r << n$. The corresponding optimization problem is solved using the Alternating Direction Method of Multipliers (ADMM).

II. Background

A. Notations

We represent a matrix of size $n \times n$ as $\mathbf{A} \in \mathbf{R}^{n\times n}$, where the (i, j)-th entry of the matrix $\mathbf{A}$ is denoted as $A_{ij} \forall i, j$. The trace of $\mathbf{A}$ is defined as: $\operatorname{tr}(\mathbf{A}) = \sum_i A_{ii}$, which is the sum of its diagonal elements. The symbol $\top$ is used to denote the transpose operation. The Frobenius norm, denoted as $\|\mathbf{A}\|_F$, is given by: $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$, which computes the square root of the sum of the squared entries of the matrix. The $\ell_{2,1}$ norm, denoted as $\|\mathbf{A}\|_{2,1}$, represents the sum of the ℓ_2-norms of the rows of $\mathbf{A}$. It is defined as: $\|\mathbf{A}\|_{2,1} = \sum_i \sqrt{\sum_j A_{i,j}^2}$. The symbol $\odot$ is the Hadamard product (element-wise) product of two matrices.

B. Graph Theory

An undirected graph $G = (V, E)$ consists of a set of n nodes in V, where $|V| = n$, and a set of edges $E \subseteq V \times V$. An edge connecting nodes i and j is denoted as E_{ij} and is associated with a weight w_{ij}. The graph G can be represented algebraically by an $n \times n$ symmetric adjacency matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$. The graph Laplacian $\mathbf{L}$ is given by: $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix called the degree matrix. Each diagonal element D_{ii} is $D_{ii} = \sum_{j=1}^{n} W_{ij}$. The eigendecomposition of $\mathbf{L}$ is $\mathbf{L} = \mathbf{U}^\top \mathbf{\Lambda} \mathbf{U}$, where $\mathbf{U}$ is a matrix whose columns are the eigenvectors of $\mathbf{L}$, and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues in ascending order, $0 = \Lambda_{11} \leq \Lambda_{22} \leq \cdots \leq \Lambda_{nn}$.

C. Smoothness Based on Graph Learning

A graph signal defined on G is a function $x : V \to \mathbb{R}$, which can be represented as a vector $\mathbf{x} \in \mathbb{R}^n$, where each entry x_i corresponds to the signal value at node i. The eigenvectors and eigenvalues of the Laplacian matrix of G are used to define the Graph Fourier Transform (GFT). The GFT of $\mathbf{x}$ is given by: $\widehat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$, where $\widehat{x}_i$ represents the Fourier coefficient at the i-th frequency component Λ_{ii}.

A graph signal $\mathbf{x}$ is considered smooth if most of the energy of $\widehat{\mathbf{x}}$ is concentrated in the low-frequency components. The smoothness of $\mathbf{x}$ is quantified by the total variation of $\mathbf{x}$, measured using the spectral density of its Fourier transform as:

$$\text{trace}(\widehat{\mathbf{x}}^\top \mathbf{\Lambda} \widehat{\mathbf{x}}) = \text{trace}(\mathbf{x}^\top \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top \mathbf{x}) = \text{trace}(\mathbf{x}^\top \mathbf{L} \mathbf{x}). \quad (1)$$

III. Multiview Graph Learning Based on Node Perturbation Model (MVGL-NP)

Given a set of signal samples for each view $\mathbf{X}^{(k)} = [\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_n^{(k)}]$, where n and k represent the number of nodes and views, respectively, the goal is to learn the individual graph structures, i.e., the graph Laplacians, $\mathbf{L}^k$. Assuming that the individual views differ due to particular nodes that are perturbed across the views, thus have a completely different connectivity pattern, the problem of learning the individual graph Laplacians, $\mathbf{L}^k$, with the smoothness assumption can be expressed as:

$$\begin{aligned} \min_{\mathbf{L}^k, \mathbf{V}^k} & \sum_{k=1}^{K} tr(\mathbf{X}^{k\top} \mathbf{L}^k \mathbf{X}^k) + \gamma_1 \|\mathbf{L}^k - \mathbf{I} \odot \mathbf{L}^k\|_F^2 \\ & - \gamma_2 tr(\log(\mathbf{I} \odot \mathbf{L}^k)) + \gamma_3 \|\mathbf{V}^k\|_{2,1} \\ \text{s.t.} \quad & \mathbf{L}^k \succeq 0, \mathbf{L}^k \cdot \mathbf{1} = 0, (K-1)\mathbf{L}^k - \sum_{j \neq k}^{K} \mathbf{L}^j = \mathbf{V}^k + \mathbf{V}^{k\top}, \end{aligned} \quad (2)$$

where $\mathbf{X}^k \in \mathbf{R}^{(n\times d_v)}$ is the data samples of the k^{th} view with dimension d_v. The first term quantifies the total variation of the observed signal, $\mathbf{X}^k$, with respect to the graph Laplacian $\mathbf{L}^k$. The second term enforces sparsity in the learned graphs, while the third term uses a logarithmic penalty on the degrees of the learned graphs to ensure connectivity [15]. The last term penalizes differences between the views using the row-column overlap norm (RCON) [9]. $\mathbf{L}^k$ is constrained to be in $= \{\mathbf{L} : \mathbf{L} \succeq \mathbf{0}, L_{ij} = L_{ji} \leq 0 \ \forall i \neq j, \ \mathbf{L1} = \mathbf{0}\}$, which is the set of valid Laplacians. The last constraint ensures that the difference between each view, $\mathbf{L}^k$, and every other view for $k \neq j$ has a row-column sparse structure captured by $\mathbf{V}^k + (\mathbf{V}^k)^\top$. In general, the optimization problem in Eq. (2) is nonconvex due to the constraints. To deal with these nonconvex constraints, we present an equivalent form of the constraints $\mathbf{L}^k \succeq 0, \mathbf{L}^k \cdot \mathbf{1} = 0$ as follows [16]:

$$\mathbf{L}^k \succeq 0, \mathbf{L}^k \cdot \mathbf{1} = 0 \iff \mathbf{P}\mathcal{E}^k\mathbf{P}^\top, \mathcal{E}^k \succeq 0, \quad (3)$$

where $\mathbf{P} \in \mathbf{R}^{n\times(n-1)}$ is the orthogonal complement of the vector $\mathbf{1}$, i.e., $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$ and $\mathbf{P}^\top\mathbf{1} = \mathbf{0}$, and $\mathcal{E}^k \in \mathbf{R}^{(n-1)\times(n-1)}$ is a positive semi-definite matrix for the k^{th} view.

The optimization problem in Eq.(2) can be solved using Alternating Direction Method of Multipliers (ADMM). The solution of the optimization problem introduced in Eq.(2) starts with introducing an auxiliary variable, $\mathbf{C}^k$, $\mathbf{W}^k$, $\mathbf{H}^k$, $\mathbf{Z}^k$, and $\mathbf{Q}^k$, to separate the variables where the optimization problem will be reformulated as follows:

$$\begin{aligned} \min_{\substack{\mathbf{V}^k, \mathbf{W}^k C^k, \mathcal{E}^k, \\ \mathbf{Z}^k, \mathbf{Q}^k, \mathbf{H}^k}} & \sum_{k=1}^{K} tr(\mathbf{B}^k\mathcal{E}^k) + \gamma_1 \|\mathbf{P}\mathcal{E}^k\mathbf{P}^\top - \mathbf{Z}^k\|_F^2 \\ & - \gamma_2 tr(\log(\mathbf{Z}^k)) + \gamma_3 \|\mathbf{Q}^k\|_{2,1} \\ \text{s.t.} \quad & \mathbf{C}^k = \mathbf{P}\mathcal{E}^k\mathbf{P}^\top, \mathbf{H}^k = \mathbf{C}^k, \mathbf{V}^k = \mathbf{Q}^k, \mathbf{I} \odot \mathbf{C}^k = \mathbf{Z}^k, \\ & (K-1)\mathbf{H}^k - \sum_{j\neq k}^{K} \mathbf{H}^j = \mathbf{V}^k + \mathbf{W}^{k\top}, \mathbf{V}^k = \mathbf{W}^{k\top}, \end{aligned} \quad (4)$$

where $\mathbf{B}^k = \mathbf{P}^\top \mathbf{X}^k \mathbf{X}^{k\top} \mathbf{P}$. The augmented Lagrangian function of Eq.(4) can be written as follows:

$$\begin{aligned} \min_{\substack{\mathbf{V}^k, \mathbf{W}^k C^k, \mathcal{E}^k, \\ \mathbf{Z}^k, \mathbf{Q}^k, \mathbf{H}^k}} & \sum_{k=1}^{K} tr(\mathbf{B}^k\mathcal{E}^k) + \gamma_1 \|\mathbf{P}\mathcal{E}^k\mathbf{P}^\top - \mathbf{Z}^k\|_F^2 - \gamma_2 tr(\log(\mathbf{Z}^k)) \\ & + \gamma_3 \|\mathbf{Q}^k\|_{2,1} + \frac{\alpha}{2} \|(K-1)\mathbf{H}^k - \sum_{j\neq k}^{K} \mathbf{H}^j - (\mathbf{V}^k + \mathbf{W}^{\top k}) + \frac{\mathbf{F}^k}{\alpha}\|_F^2 \\ & + \frac{\alpha}{2} \|\mathbf{V}^k - \mathbf{W}^{\top k} + \frac{\mathbf{G}^k}{\alpha}\|_F^2 + \frac{\alpha}{2} \|\mathbf{H}^k - \mathbf{C}^k + \frac{\mathbf{M}^k}{\alpha}\|_F^2 \\ & + \frac{\alpha}{2} \|\mathbf{Z}^k - \mathbf{I} \odot \mathbf{C}^k + \frac{\mathbf{U}^k}{\alpha}\|_F^2 + \frac{\alpha}{2} \|\mathbf{C}^k - \mathbf{P}\mathcal{E}^k\mathbf{P}^\top + \frac{\mathbf{Y}^k}{\alpha}\|_F^2 \\ & + \frac{\alpha}{2} \|\mathbf{V}^k - \mathbf{Q}^k + \frac{\mathbf{N}^k}{\alpha}\|_F^2, \end{aligned} \quad (5)$$

where $\mathbf{Y}^k$, $\mathbf{F}^k$, $\mathbf{G}^k$, $\mathbf{U}^k$ $\mathbf{M}^k$, and $\mathbf{N}^k$ are the Lagrangian multipliers and α is the penalty parameter. Eq. (5) can be solved by dividing it into multiple subproblems and optimizing each variable while fixing the others as follows:

- **Subproblem** $\mathcal{E}^k$: To update $\mathcal{E}^k$, we fix all the other variables and consider only the terms with $\mathcal{E}^k$ as follows:

$$\mathcal{E}_{l+1}^k = \min_{\mathcal{E}^k} \sum_{k=1}^{K} \Big[tr(\mathbf{B}^k \mathcal{E}_l^k) + \gamma_1 \|\mathbf{P}\mathcal{E}^k\mathbf{P}^\top - \mathbf{Z}^k\|_F^2 + \frac{\alpha}{2}\|\mathbf{C}_l^k - \mathbf{P}\mathcal{E}_l^k\mathbf{P}^\top + \frac{\mathbf{Y}_l^k}{\alpha}\|_F^2 \Big]. \tag{6}$$

By taking the gradient with respect to $\mathcal{E}^k$ and equating it to zero, the solution of Eq. (6) can be found as follows:

$$\mathcal{E}_{l+1}^k = \frac{2\gamma_1 \mathbf{P}^\top \mathbf{Z}^k \mathbf{P} + \alpha \mathbf{P}^\top \mathbf{C}_l^k \mathbf{P} + \mathbf{P}^\top \mathbf{Y}_l^k \mathbf{P} - {\mathbf{B}^k}^\top}{2\gamma_1 + \alpha}. \tag{7}$$

- **Subproblem** $\mathbf{Z}^k$: The solution of $\mathbf{Z}^k$ can be found by solving the following problem:

$$\mathbf{Z}_{l+1}^k = \min_{\mathbf{Z}^k} \sum_{k=1}^{K} \Big[\gamma_1 \|\mathbf{P}\mathcal{E}^k\mathbf{P}^\top - \mathbf{Z}^k\|_F^2 - \gamma_2 tr(\log(\mathbf{Z}^k)) + \frac{\alpha}{2}\|\mathbf{Z}^k - \mathbf{I}\odot\mathbf{C}^k + \frac{\mathbf{U}^k}{\alpha}\|_F^2 \Big]. \tag{8}$$

The solution of Eq. 8 can be found as :

$$\mathbf{Z}_{l+1}^k = \frac{\mathbf{J}^k + \sqrt{(\mathbf{J}^k)^2 + 4(2\gamma_1+\alpha)\gamma_2\mathbf{I}}}{2(2\gamma_1+\alpha)}, \tag{9}$$

where $\mathbf{J}^k = 2\gamma_1 \mathbf{P}\mathcal{E}^k\mathbf{P}^\top + \alpha\mathbf{I}\odot\mathbf{C}^k - \mathbf{U}^k$.

- **Subproblem** $\mathbf{C}^k$: The solution of $\mathbf{C}^k$ can be found by solving the following problem:

$$\mathbf{C}_{l+1}^k = \min_{\mathbf{C}^k} \sum_{k=1}^{K} \Big[\frac{\alpha}{2}\|\mathbf{C}_l^k - \mathbf{P}\mathcal{E}_{l+1}^k\mathbf{P}^\top + \frac{\mathbf{Y}^k}{\alpha}\|_F^2 + \frac{\alpha}{2}\|\mathbf{H}^k - \mathbf{C}^k + \frac{\mathbf{M}^k}{\alpha}\|_F^2 + \|\mathbf{Z}^k - \mathbf{I}\odot\mathbf{C}^k + \frac{\mathbf{U}^k}{\alpha}\|_F^2 \Big]. \tag{10}$$

The problem in (10) will update the diagonal and off-diagonal parts of the matrices, separately. To find the solution, the gradient with respect to $\mathbf{C}^{(k)}$ is taken and set to zero. The resulting solution can be expressed as follows:

$$[\mathbf{C}_{l+1}^k]_{ij} = \begin{cases} \frac{\alpha[\mathbf{P}\mathcal{E}^k\mathbf{P}^\top]_{ii} - Y_{ii}^k + \alpha H_{ii}^k + M_{ii}^k + Z_{ii}^k + U_{ii}^k}{3\alpha}, & \text{for} \quad i = j, \\ \frac{\alpha[\mathbf{P}\mathcal{E}^k\mathbf{P}^\top]_{ij} - Y_{ij}^k + \alpha H_{ij}^k + M_{ij}^k}{2\alpha}, & \text{for} \quad i \neq j. \end{cases} \tag{11}$$

- **Subproblem** $\mathbf{H}^k$: In order to update $\mathbf{H}^k$, we fix all the other variables and consider the terms with $\mathbf{H}^k$ only as follows:

$$\mathbf{H}_{l+1}^k = \min_{\mathbf{H}^k} \sum_{k=1}^{K} \Big[\frac{\alpha}{2}\|\mathbf{H}_l^k - \mathbf{C}_{l+1}^k + \frac{\mathbf{M}_l^k}{\alpha}\|_F^2 + \frac{\alpha}{2}\|(K-1)\mathbf{H}_l^k - \sum_{j\neq k}^{K}\mathbf{H}_l^j - (\mathbf{V}_l^k + \mathbf{W}_l^k) + \frac{\mathbf{F}_l^k}{\alpha}\|_F^2 \Big]. \tag{12}$$

By taking the gradient of (12) and setting it to zero, the solution of $\mathbf{H}^k$ can be written as follows:

$$[\mathbf{H}_{l+1}^k]_{ij} = \begin{cases} \Big[\Gamma_h^k \odot \mathbf{I}\Big]_+, & \text{for} \quad i = j \\ \Big[\Gamma_h^k \odot (\mathbf{1}\mathbf{1}^\top - \mathbf{I})\Big]_-, & \text{for} \quad i \neq j, \end{cases} \tag{13}$$

where $\Gamma_h^k = \frac{\alpha\sum_{j\neq k}^{K} H_l^j + \alpha\mathbf{V}_l^k + \alpha\mathbf{W}_l^k - \mathbf{F}_l^k + \alpha\mathbf{C}_{l+1}^k - \mathbf{M}_l^k}{\alpha K}$. The term $\Big[\Gamma_h^k \odot \mathbf{I}\Big]_+$ represents a diagonal matrix in which all elements are non-negative, where positive elements are retained and negative elements are set to zero. The term $\Big[\Gamma_h^k \odot (\mathbf{1}\mathbf{1}^\top - \mathbf{I})\Big]_-$ represents an off-diagonal matrix where only negative elements are kept, and positive elements are replaced by zeros. This ensures that $\mathbf{H}^k$ is a valid Laplacian matrix.

- **Subproblem** $\mathbf{V}^k$: In order to update $\mathbf{V}^k$, we fix all the other variables and consider the terms with $\mathbf{V}^k$ only as follows:

$$\mathbf{V}_{l+1}^k = \min_{\mathbf{V}^k} \sum_{k=1}^{K} \Big[\frac{\alpha}{2}\|\mathbf{V}_l^k - {\mathbf{W}_l^k}^\top + \frac{\mathbf{G}_l^k}{\alpha}\|_F^2 + \frac{\alpha}{2}\|\mathbf{Q}_l^k - \mathbf{V}_l^k + \frac{\mathbf{N}_l^k}{\alpha}\|_F^2 + \frac{\alpha}{2}\|(K-1)\mathbf{H}_{l+1}^k - \sum_{j\neq k}^{K}\mathbf{H}_{l+1}^j - (\mathbf{V}_l^k + \mathbf{W}_l^k) + \frac{\mathbf{F}_l^k}{\alpha}\|_F^2 \Big]. \tag{14}$$

By taking the derivative and equating it to zero, the solution of Eq. (14) can be found as:

$$\mathbf{V}_{l+1}^k = \frac{\phi^k - \alpha\mathbf{W}_l^k + \mathbf{F}_l^k + \alpha\mathbf{Q}_l^k + \mathbf{N}_l^k + \alpha{\mathbf{W}_l^k}^\top - \mathbf{G}_l^k}{3\alpha}, \tag{15}$$

where $\phi^k = \alpha(K-1)\mathbf{H}_{l+1}^k - \alpha\sum_{j\neq k}^{K}\mathbf{H}_{l+1}^j$.

- **Subproblem** $\mathbf{Q}^k$: In order to update $\mathbf{Q}^k$, we fix all the other variables and consider the terms with $\mathbf{Q}^k$ only as follows:

$$\mathbf{Q}_{l+1}^k = \min_{\mathbf{Q}^k} \gamma_3\|\mathbf{Q}_l^k\|_{2,1} + \frac{\alpha}{2}\|\mathbf{Q}_l^k - \mathbf{V}_{l+1}^k + \frac{\mathbf{N}_l^k}{\alpha}\|_F^2. \tag{16}$$

The solution of Eq. (16) can be found by using the proximal operator for $l_{2,1}$-norm, and it can be found as follows:

$$\mathbf{Q}_{l+1}^k = \mathcal{T}_{2,1}(\mathbf{V}_{l+1} + \frac{\mathbf{N}_l}{\alpha}, \frac{\gamma_3}{\alpha}). \tag{17}$$

- **Subproblem** $\mathbf{W}^k$: The solution of $\mathbf{W}_{l+1}^k$ can be found by solving the following minimization problem:

$$\mathbf{W}_{l+1}^k = \min_{\mathbf{W}^k} \sum_{k=1}^{K} \Big[\frac{\alpha}{2}\|(K-1)\mathbf{H}_{l+1}^k - \sum_{j\neq k}^{K}\mathbf{H}_{l+1}^j - (\mathbf{V}_{l+1}^k + \mathbf{W}_l^k) + \frac{\mathbf{F}_l^k}{\alpha}\|_F^2 + \frac{\alpha}{2}\|\mathbf{V}_{l+1}^k - {\mathbf{W}_l^k}^\top + \frac{\mathbf{G}_l^k}{\alpha}\|_F^2 \Big]. \tag{18}$$

Then, the solution of $\mathbf{W}_{l+1}$ can be written as follows:

$$\mathbf{W}_{l+1}^k = \frac{\mathbf{\Psi}^k - \alpha\mathbf{V}_{l+1}^k + \mathbf{F}_l^k + \alpha\mathbf{V}_k^\top{}_{l+1} + \mathbf{G}_l^\top}{2\alpha}, \tag{19}$$

where $\mathbf{\Psi}^k = \alpha(K-1)\mathbf{H}^k_{l+1} - \alpha \sum_{j \neq k}^{K} \mathbf{H}^j_{l+1}$.
Finally, the Lagrangian multipliers and the penalty parameters can be updated as follows:

$$
\begin{aligned}
&\mathbf{Y}^k_{l+1} = \mathbf{Y}^k_l + \alpha(\mathbf{C}^k_{l+1} - \mathbf{P}\mathcal{E}^k_{l+1}\mathbf{P}^\top).\\
&\mathbf{F}^k_{l+1} = \mathbf{F}^k_l + \alpha((K-1)\mathbf{H}^k_{l+1} - \sum_{j \neq k}^{K} \mathbf{H}^j_{l+1} - (\mathbf{V}^k_{l+1} + \mathbf{W}^k_{l+1})).\\
&\mathbf{G}^k_{l+1} = \mathbf{G}^k_l + \alpha(\mathbf{V}^k_{l+1} - {\mathbf{W}^\top}^k_{l+1}).\\
&\mathbf{M}^k_{l+1} = \mathbf{M}^k_l + \alpha(\mathbf{H}^k_{l+1} - \mathbf{C}^k_{l+1}).\\
&\mathbf{N}^k_{l+1} = \mathbf{N}^k_l + \alpha(\mathbf{V}^k_{l+1} - \mathbf{Q}^k_{l+1}).\\
&\mathbf{U}^k_{l+1} = \mathbf{U}^k_l + \alpha(\mathbf{Z}^k_{l+1} - \mathbf{I} \odot \mathbf{C}^k_{l+1}).\\
&\alpha_{l+1} = \mu\alpha_l, \mu > 1.
\end{aligned}
\tag{20}
$$

IV. Experimental Results

A. Simulated Networks

In our experiments, an Erdős-Rényi (ER) random network model is used to test the performance of the proposed model. Node pairs are connected independently with a fixed probability of 0.1, resulting in a graph where edges are distributed uniformly at random. In this network model, the adjacency matrix $\mathbf{A}$ is replicated K times to generate the different views. To introduce perturbations, r nodes are randomly selected. For each of these randomly selected nodes, the corresponding rows and columns in all K matrices are updated with entries independently sampled from a Bernoulli distribution. This process generates r perturbed nodes across K views.

B. Data Generation

Given K view graphs, each $\mathbf{X}^k \in \mathbb{R}^{n \times d_k}$ is derived from G^k using the smooth graph filter $h(\mathbf{L}^k)$ as described in [17]. Specifically, each column of $\mathbf{X}^k$ is generated as $\mathbf{X}^k_{\cdot j} = h(\mathbf{L}^k)x_0$, where $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. In this paper, we used a Gaussian filter ($h(\mathbf{L}) = \mathbf{L}^\dagger$), which results in Gaussian distributed signals where the graph Laplacian and the precision matrix are equivalent. Additionally, $\eta\%$ noise is added to each generated $\mathbf{X}^k$. The average F-score is reported across 10 realizations.

C. Benchmark Models

We evaluate the proposed method by comparing it with the following approaches:

- SV: A single-view graph learning method proposed in [1], which independently learns the graph topology for each view under the assumption that the signals are smooth with respect to the graph of each view.
- MVGL: A multiview graph learning approach introduced in [14], which jointly learns multiple graph Laplacians by assuming signal smoothness on each view's graph. Additionally, it enforces similarity between views by minimizing the ℓ_1-norm error between each view's graph and the learned consensus graph.
- GGL: Group Graphical Lasso (GGL) model utilizes group lasso penalties to promote topological similarity across different views. However, its effectiveness is constrained by the assumption that the observed graph signals follow a Gaussian distribution, which is often not the case in real-world networks [5].

1) Results and Discussion: In the first experiment, we fix the number of nodes, $n = 100$, the number of signal samples, $d_v = 900$, the number of perturbed nodes, $r = 4$, and vary the number of views. As it can be seen in Fig. 1, as the number of views increases the performance of multiview graph learning techniques improve while the performance of single view method stays constant. The proposed method outperforms both multiview Laplacian learning and GGL as it takes the perturbation model into account.

In the second experiment, we fix the number of views, $K = 8$, the number of nodes and perturbed nodes to, $n = 100$, $r = 4$, respectively, and vary the number of signal samples. The performance of all methods improve with the increase in the number of samples.

Finally, in the last experiment, we fix the number of nodes, $n = 100$, the number of samples $d_v = 900$, the number of views, $K = 10$, and vary the number of perturbed nodes. As more nodes are perturbed, the assumption of the difference between views having a few number of nonzero rows and columns becomes weaker thus reducing the performance of all methods. However, MVGL-NP is still the most robust one against perturbations.

V. Conclusions

In this paper, we introduces a node-based multiview graph learning framework. The proposed framework assumes the smoothness of graph signals with respect to the underlying graph structure while making the assumption that each view is a perturbed version of the others through a random wiring of a small number of nodes' connections. Based on this assumption, we introduced an optimization framework that incorporated the structured sparsity of the difference between pairs of views. The proposed framework is evaluated on simulated graph signal models for different number of signals, views and perturbed nodes. Future work will consider other node-based similarity metrics across views such as the co-hub model.

References

[1] 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, 2016.

[2] 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, 2019.

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

[4] K. Tsai, O. Koyejo, and M. Kolar, "Joint gaussian graphical model estimation: A survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 14, no. 6, p. e1582, 2022.

[5] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 76, no. 2, pp. 373–397, 2014.

[6] M. Navarro, Y. Wang, A. G. Marques, C. Uhler, and S. Segarra, "Joint inference of multiple graphs from matrix polynomials," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 3302–3336, 2022.

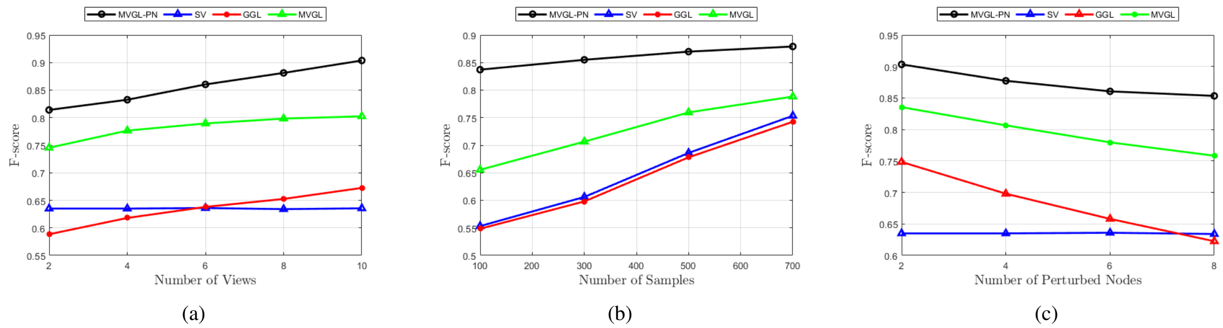

Fig. 1. The impact of increasing the number of (a) views, (b) samples,and (c) perturbed nodes on F-score value.

[7] J. Guo, E. Levina, G. Michailidis, and J. Zhu, "Joint estimation of multiple graphical models," *Biometrika*, vol. 98, no. 1, pp. 1–15, 2011.
[8] W. Lee and Y. Liu, "Joint estimation of multiple precision matrices with common structures," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1035–1062, 2015.
[9] K. Mohan, P. London, M. Fazel, D. Witten, and S.-I. Lee, "Node-based learning of multiple gaussian graphical models," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 445–488, 2014.
[10] J. Ma and G. Michailidis, "Joint structural estimation of multiple graphical models," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 5777–5824, 2016.
[11] F. Huang and S. Chen, "Joint learning of multiple sparse matrix gaussian graphical models," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 11, pp. 2606–2620, 2015.
[12] Y. Yuan, K. Guo, Z. Xiong, T. Q. Quek *et al.*, "Joint network topology inference via structural fusion regularization," *IEEE Transactions on Knowledge and Data Engineering*, 2023.
[13] X. Zhang and Q. Wang, "A graph-assisted framework for multiple graph learning," *IEEE Transactions on Signal and Information Processing over Networks*, 2024.
[14] A. Karaaslanli and S. Aviyente, "Multiview graph learning with consensus graph," *arXiv preprint arXiv:2401.13769*, 2024.
[15] 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)*. Ieee, 2017, pp. 2826–2830.
[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, 2019.
[17] V. Kalofolias, "How to learn a graph from smooth signals," in *Artificial intelligence and statistics*. PMLR, 2016, pp. 920–929.