Learning Cartesian Product Graphs with Laplacian Constraints

Changhao Shi **Gal Mishne**
University of California San Diego

Abstract

Graph Laplacian learning, also known as network topology inference, is a problem of great interest to multiple communities. In Gaussian graphical models (GM), graph learning amounts to endowing covariance selection with the Laplacian structure. In graph signal processing (GSP), it is essential to infer the unobserved graph from the outputs of a filtering system. In this paper, we study the problem of learning Cartesian product graphs under Laplacian constraints. The Cartesian graph product is a natural way for modeling higher-order conditional dependencies and is also the key for generalizing GSP to multi-way tensors. We establish statistical consistency for the penalized maximum likelihood estimation (MLE) of a Cartesian product Laplacian, and propose an efficient algorithm to solve the problem. We also extend our method for efficient joint graph learning and imputation in the presence of structural missing values. Experiments on synthetic and real-world datasets demonstrate that our method is superior to previous GSP and GM methods.

1 INTRODUCTION

Graphs are powerful tools for modeling relationships among a set of entities in complex systems and have become prevalent in biology (Pavlopoulos et al., 2011), neuroscience (Bassett and Sporns, 2017), social science (Borgatti et al., 2009), and many other scientific fields. In machine learning and artificial intelligence, there is also a growing interest in graphs for model boosting (Shuman et al., 2013; Wu et al., 2020; Ji et al., 2021). As the graph of a system is frequently unobserved, a prominent problem in graph machine learning is how to construct a graph from available data for further use. While ad-hoc graph construction rules (e.g. k-nearest neighbor graphs) exist in many fields, it is arguably more appealing to learn those graphs in a more principled way. To be more specific, given a set of nodes and some nodal observations attached to them, we aim to infer their edge connectivity pattern. In the literature, this problem is termed graph learning or network topology inference (Mateos et al., 2019).

Graph learning is a central problem in GSP, the subject that generalizes traditional signal processing (SP) to non-Euclidean domains (Shuman et al., 2013; Ortega et al., 2018). In analog to traditional SP, GSP uses eigenfunctions of various graph representations, such as adjacency and Laplacian matrices, to define a graph Fourier basis that transforms graph signals to the spectral domain, where frequency analysis and filtering can be applied. If assuming nodal observations should be smooth with respect to the true unseen graph (usually sparse), we can define a class of graph learning methods favoring the "smoothness prior". This boils down to minimizing the total variation of nodal observations with respect to the combinatorial graph Laplacian.

Interestingly, this graph learning formulation is closely related to covariance selection in GM (Dempster, 1972). Covariance selection aims to estimate a sparse inverse covariance matrix, or a precision matrix, from a sample covariance matrix (SCM). Enforcing Laplacian structure on the precision matrix (and considering its pseudo-inverse) will endow covariance selection with a similar form to the smoothness prior. The resultant problem is essentially the penalized MLE of an attractive, improper Gaussian Markov random field (IGMRF), which interests the GM community.

While graph Laplacian learning and covariance selection are useful for single-way analysis, they are not intended for multi-way tensors. A multi-way tensor, as opposed to a single-way vector, is a multi-dimensional array where each way or mode of the tensor represents a different source of variation. Consider such

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

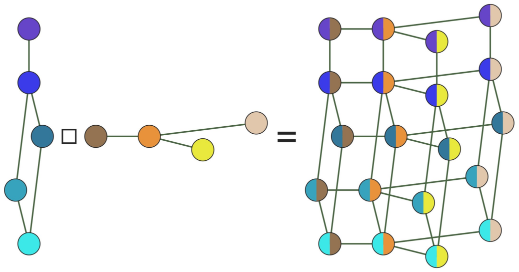


Figure 1: An example of the Cartesian graph product.

a multi-way scenario: a sensor network of p_s sensors with unknown connectivity and their measurements on p_t time points over a day. An example of single-way inference is to directly apply graph learning methods to learn a graph of sensors from the p_t 1-d spatial signals. Not surprisingly, this results in a sub-optimal solution since the dependencies between p_t time points are ignored. A more appealing approach is to learn a graph of size $p_s p_t$, in which each node is a (sensor, time point) pair. However, this poses new computational challenges since $p_s p_t$ is usually huge.

To circumvent the challenges, imposing graphs with the Cartesian product structure has gained massive interest Sandryhaila and Moura (2014); Zhang et al. (2021); He et al. (2023). An example of the Cartesian graph product is shown in Fig. 1. As we can see, Cartesian product graphs are extremely suitable for multi-way data, since they offer a reasonable parsimony where only dependencies within ways are captured by factor graphs. It is even more appealing to learn Cartesian product graphs under the Laplacian constraints, which serve as the foundations of multi-way GSP (Stanley et al., 2020). Owing to the Cartesian product Laplacian, the multi-way graph Fourier transform enjoys a concise form that separates to mode-wise Fourier transform.

In this paper, we study the problem of learning the Cartesian product Laplacian from multi-way data. We consider the penalized MLE of the Cartesian product IGMRF and propose an efficient algorithm to solve the problem by leveraging the spectral properties of the Cartesian product Laplacian. A modified algorithm is also proposed for joint graph learning and missing value imputation. On the theoretical aspect, we establish the high-dimensional statistical consistency of the proposed penalized MLE and obtain an improved rate of convergence over non-product graph Laplacian learning. Our method provides a better solution than related GM works, which ignore the Laplacian constraints, and existing GSP works, which lack theoretical guarantees. To summarize our contributions:

- We are the first to consider the penalized MLE of Cartesian product Laplacian learning, and gain theoretical results on its asymptotic consistency, to the best of our knowledge.
- We propose an efficient algorithm to solve the penalized MLE, which reduces the time complexity of the naive solution. We further extend the algorithm to the setting of structural missing data.
- We demonstrate that our approach outperforms existing GSP and GM methods on synthetic and real-world datasets.

As a side note, we emphasize that graph learning is intrinsically a different problem from covariance selection, although they bear a similar form. The parameter space of graph learning and covariance selection are two disjoint sets as Laplacian matrices are singular with constant 0-eigenvectors. Graph learning also requires that all conditional dependencies are positive, though this is also a GM subject under the study of M-matrices (Slawski and Hein, 2015). Our code is released at `https://github.com/Mishne-Lab/MWGL`.

1.1 Related Work

In **GSP**, the smoothness prior is arguably the most common method for graph learning (Dong et al., 2016; Kalofolias, 2016; Chepuri et al., 2017; Egilmez et al., 2017; Zhao et al., 2019; Buciulea et al., 2022; Shi and Mishne, 2023). Other Laplacian-based models, such as heat diffusion (Thanou et al., 2017), and models based on the normalized graph Laplacian matrix (Pasdeloup et al., 2017) and the weighted adjacency matrix (Segarra et al., 2017; Navarro et al., 2020; Shafipour et al., 2021) have also been studied. In terms of learning Cartesian product graphs, Lodhi and Bajwa (2020) advocated directly optimizing the total variation on product Laplacian; Kadambari and Prabhakar Chepuri (2020); Kadambari and Chepuri (2021) decomposed the overall smoothness measurement into factor-wise variation so that each factor graph can be learned separately; Einizade and Sardouie (2023) proposed to first estimate eigenfunctions of factor graph representations, and then solve the spectral template problem as in (Segarra et al., 2017). However, these methods simplified the MLE to facilitate optimization, which generally leads to asymptotic inconsistencies.

In **GM**, covariance selection (Dempster, 1972) aims to obtain a parsimonious model of the conditional dependencies, which amounts to learning a sparse precision matrix in a GMRF (Banerjee et al., 2006; Yuan and Lin, 2007; Banerjee et al., 2008). In modern days, this problem is efficiently solved by the prestigious graphical lasso algorithm and its variants (Friedman et al., 2008; d'Aspremont et al., 2008; Lu, 2009; Scheinberg et al., 2010; Li and Toh, 2010; Hsieh et al., 2011; Witten et al., 2011; Mazumder and Hastie, 2012; Oztoprak et al., 2012). Since then, the graphical lasso algorithm has been extended to matrix/tensor Gaussian distributions (Dawid, 1981; Gupta and Nagar, 1999) to learn Kronecker product precision matrices (Dutilleul, 1999; Zhang and Schneider, 2010; Leng and Tang, 2012; Tsiligkaridis et al., 2013). Further extensions replace the Kronecker product structure with the Kronecker sum (Kalaitzis et al., 2013; Greenewald et al., 2019; Wang et al., 2020; Yoon and Kim, 2022), leading to Cartesian product graphs. Again, none of these graphical lasso methods learn precision matrices under Laplacian constraints but only bear a similar form to the Cartesian product Laplacian learning.

2 BACKGROUND

We use the following notations throughout the paper. Lower-case and upper-case bold letters denote vectors and matrices respectively, and lower-case bold italic letters denote random vectors. Let $\mathbf{1}$ and $\mathbf{0}$ denote the all 1 and all 0 vectors, and let $\mathbf{O}$ denote the all 0 matrix. Let $\mathbf{e}_p^l \in \mathbb{R}^p$ denote a unit vector that has 1 in its l-th entry. $\dagger$ denotes the Moore-Penrose pseudo-inverse and $\det^\dagger$ denotes the pseudo-determinant. $\circ$ denotes the Hadamard product. $\otimes$ and $\oplus$ denote the Kronecker product and Kronecker sum. The Kronecker sum of two matrices $\mathbf{M_1}$ and $\mathbf{M_2}$ is defined as $\mathbf{M_1} \oplus \mathbf{M_2} = \mathbf{M_1} \otimes \mathbf{I_2} + \mathbf{I_1} \otimes \mathbf{M_2}$. For graphs G_1 and G_2, we use $\square$ to denote the Cartesian graph product operator. Let $\times$ denote the Cartesian product of two sets. For matrix norms, $\|\cdot\|_F$ denotes the Frobenius norm, $\|\cdot\|_2$ the operator norm, and $\|\cdot\|_{1,\text{off}}$ sum of the absolute values of all off-diagonal elements. For random variables, $\|\cdot\|_{\psi_2}$ denotes the sub-Gaussian norm. $(\cdot)_+$ denotes projection to the non-negative plane and $\mathbb{1}$ the indicator function.

2.1 Preliminaries

Let $G = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ be an undirected, connected graph where $\mathcal{V}$ denotes the set of p vertices, $\mathcal{E}$ the set of edges and $\mathbf{W} \in \mathcal{S}^p$ the weighted adjacency matrix. Each entry of the weight matrix $[\mathbf{W}]_{ij} = [\mathbf{W}]_{ji} \geq 0$ encodes the similarity between a node pair (i, j), and $[\mathbf{W}]_{ij} = [\mathbf{W}]_{ji} > 0$ iff $e_{ij} \in \mathcal{E}$. We assume there are no self-loops, i.e. $\mathbf{W}_{ii} = 0$, and denote the vectorization of all the weights as $\mathbf{w} \in \mathbb{R}^{p(p-1)/2}$, such that $[\mathbf{w}]_{i-j+\frac{1}{2}(j-1)(2n-j)} = [\mathbf{W}]_{ij}, \forall 1 \leq j \leq i \leq p$. The combinatorial graph Laplacian matrix $\mathbf{L}$ of the graph G is given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ denotes the diagonal degree matrix where $[\mathbf{D}]_{ii} = \sum_j [\mathbf{W}]_{ij}$. $\mathbf{W}$ and $\mathbf{L}$ are often referred to as graph representations, as each can fully determine the graph G.

In addition, we follow (Kumar et al., 2020) to define a linear operator that maps a non-negative weight vector to the corresponding combinatorial graph Laplacian.

Definition 2.1. *Define* $\mathcal{L} : \mathbb{R}^{p(p-1)/2} \to \mathbb{R}^{p \times p}, \mathbf{w} \mapsto \mathcal{L}\mathbf{w}$ *as the following linear operator*

$$[\mathcal{L}\mathbf{w}]_{ij} = \begin{cases} -[\mathbf{w}]_{i-j+\frac{1}{2}(j-1)(2p-j)} & i > j, \\ [\mathcal{L}\mathbf{w}]_{ji} & i < j, \\ -\sum_{k \neq j} [\mathcal{L}\mathbf{w}]_{kj} & i = j. \end{cases}$$

One can verify that $\mathcal{L}\mathbf{w}$ is a combinatorial graph Laplacian with weights $\mathbf{w}$. We then define its adjoint operator $\mathcal{L}^*$ such that $\langle \mathcal{L}\mathbf{w}, \mathbf{Q} \rangle = \langle \mathbf{w}, \mathcal{L}^*\mathbf{Q} \rangle, \forall \mathbf{Q} \in \mathbb{R}^{p \times p}$.

Definition 2.2. *Define* $\mathcal{L}^* : \mathbb{R}^{p \times p} \to \mathbb{R}^{p(p-1)/2}, \mathbf{Q} \mapsto \mathcal{L}^*\mathbf{Q}$ *as the following*

$$[\mathcal{L}^*\mathbf{Q}]_l = [\mathbf{Q}]_{ii} - [\mathbf{Q}]_{ij} - [\mathbf{Q}]_{ji} + [\mathbf{Q}]_{jj},$$

$$l = i - j + \frac{1}{2}(j-1)(2p-j),\ i > j.$$

Consider two weighted undirected graphs $G_1 = \{\mathcal{V}_1, \mathcal{E}_1, \mathbf{W}_1\}$ and $G_2 = \{\mathcal{V}_2, \mathcal{E}_2, \mathbf{W}_2\}$, with cardinality $|\mathcal{V}_1| = p_1$ and $|\mathcal{V}_2| = p_2$. The Cartesian product of them is denoted as $G = G_1 \square G_2$, where G_1 and G_2 are referred to as the factor graphs. The vertex set of G is defined as $\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2$. So each node $v \in \mathcal{V}$ is indexed by a node pair $(v_1 \in \mathcal{V}_1, v_2 \in \mathcal{V}_2)$, and the cardinality of G is $p = p_1 p_2$. For a node pair (v_1, v_2) and (u_1, u_2) in the product graph G, $(v_1, v_2) \sim (u_1, u_2)$ holds iff $v_1 = v_2 \wedge u_1 \sim u_2$ or $v_1 \sim v_2 \wedge u_1 = u_2$. The weighted adjacency matrix of G is the Kronecker sum of the factor weights $\mathbf{W} = \mathbf{W}_1 \oplus \mathbf{W}_2$, and similarly for the Laplacian $\mathbf{L} = \mathbf{L}_1 \oplus \mathbf{L}_2$.

2.2 Smoothness Prior

Formally, let a graph signal be a random variable $\boldsymbol{f} : \mathcal{V} \to \mathbb{R}^p$ that assigns a real value to each vertex of the graph. Let $\mathbf{f}$ be an instantiation of the graph signal. The Laplacian quadratic form, also known as the Dirichlet energy of $\mathbf{f}$, is defined as $\mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{ij} [\mathbf{W}]_{ij}([\mathbf{f}]_i - [\mathbf{f}]_j)^2$, which measures the smoothness (variation) of $\mathbf{f}$ with respect to G. Given n graph signals (instantiations) and their SCM $\mathbf{S} = \frac{1}{n}\sum_{k=1}^n \mathbf{f}_k\mathbf{f}_k^T$, $\mathcal{J}(\{\mathbf{f}_k\}) := \operatorname{Tr}(\mathbf{LS})$ measures the overall smoothness of these signals with respect to the graph. GSP seeks to learn the Laplacian $\mathbf{L}$ by solving the regularized smoothness problem

$$\min_{\mathbf{L} \in \Omega_\mathbf{L}} \left\{ \operatorname{Tr}(\mathbf{LS}) + \alpha h(\mathbf{L}) \right\}, \tag{1}$$

where $\Omega_\mathbf{L}$ is the set of all combinatorial graph Laplacian matrices

$$\Omega_\mathbf{L} := \left\{ \mathbf{L} \in \mathbf{S}_+^p \mid \mathbf{L1} = \mathbf{0}, [\mathbf{L}]_{ij} = [\mathbf{L}]_{ji} \leq 0, \forall i \neq j \right\},$$

$h(\mathbf{L})$ is a regularization term, and $\alpha > 0$ is a regularization parameter. Minimizing $\mathcal{J}(\{\mathbf{f}_k\})$ encourages the signals $\{\mathbf{f}_k\}$ to vary smoothly on the inferred graph $\mathbf{L}$. The regularization $h(\mathbf{L})$ encodes structural priors such as sparsity, and more importantly it prevents degenerate solutions such as a set of p isolated nodes resulting in $\mathbf{L} = \mathbf{O}$.

From the perspective of GM, let an IGMRF be $\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^\dagger)$, the penalized MLE gives an estimation of $\mathbf{L}$

$$\min_{\mathbf{L}\in\Omega_\mathbf{L}} \left\{ \mathrm{Tr}(\mathbf{LS}) - \log\det{}^\dagger(\mathbf{L}) + \alpha\|\mathbf{L}\|_{1,\text{off}} \right\}. \quad (2)$$

The additional ℓ_1 regularization promotes sparsity, and $\alpha > 0$ controls its strength. The penalized MLE is almost a standard covariance selection problem, with the only difference being that the precision matrix is constrained to be a combinatorial graph Laplacian.

We notice the similarity between (1) and (2). The IGMRF formulation (2) can be interpreted as a particular case of the GSP formulation (1), where $h(\mathbf{L}) = -\log\det^\dagger(\mathbf{L}) + \alpha\|\mathbf{L}\|_{1,\text{off}}$. To further see the connection from IGMRF to GSP, consider a system $\boldsymbol{f} = \mathcal{F}(\mathbf{L})\boldsymbol{f}_0$ where $\mathcal{F}(\mathbf{L})$ is the graph filter. Let the input signals be random Gaussian noise $\boldsymbol{f}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, and let the graph filter be a low-pass one $\mathcal{F}(\mathbf{L}) = \sqrt{\mathbf{L}^\dagger} = \mathbf{U}\sqrt{\mathbf{\Lambda}^\dagger}\mathbf{U}^T$, where $\mathbf{L} = \mathbf{U\Lambda U}^T$ is the eigendecomposition. This leads to the same $\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^\dagger)$. Thus, fitting this IGMRF is equivalent to estimating the graph filter of the given form under previous assumptions.

3 PRODUCT GRAPH LEARNING

3.1 Penalized MLE

Let the random matrix $\boldsymbol{X} \in \mathbb{R}^{p_1\times p_2}$ represent a two-way graph signal that lives on the product graph G. $[\boldsymbol{X}]_{i_1,i_2}$ is the signal on node (i_1, i_2). Given n instantiations $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$, our goal is to learn the factor graphs G_1, G_2 and their Cartesian product G from these nodal observations on G. Note that to ease the presentation, we limit the number of factor graphs to two, but our formulation and solution can be easily generalized to more factors and higher-dimensional tensors $\boldsymbol{X} \in \mathbb{R}^{p_1\times p_2\times p_3\times\cdots}$.

Let the random vector $\boldsymbol{x}$ be the vectorization of $\boldsymbol{X}$ and $\mathbf{S} = \frac{1}{n}\sum_{k=1}^n \mathbf{x}_k\mathbf{x}_k{}^T$ be the SCM. Since for $G = G_1\Box G_2$ we have $\mathbf{L} = \mathbf{L}_1 \oplus \mathbf{L}_2$, we derive the product graph learning objective

$$\min_{\mathbf{L}_1,\mathbf{L}_2\in\Omega_\mathbf{L}} \Big\{ \mathrm{Tr}((\mathbf{L}_1\oplus\mathbf{L}_2)\mathbf{S}) - \log\det{}^\dagger(\mathbf{L}_1\oplus\mathbf{L}_2) + \alpha\|\mathbf{L}_1\oplus\mathbf{L}_2\|_{1,\text{off}} \Big\}. \quad (3)$$

Similar to (2), (3) can be interpreted from either a GSP or a GM perspective, which we discuss below.

GSP Interpretation: We decompose $\mathcal{J}(\{\mathbf{X}_k\}) = \mathrm{Tr}((\mathbf{L}_1\oplus\mathbf{L}_2)\mathbf{S}) = \mathrm{Tr}(\mathbf{L}_1\mathbf{S}_1) + \mathrm{Tr}(\mathbf{L}_2\mathbf{S}_2)$, where

$$\mathbf{S}_1 = \frac{1}{n}\sum_{k=1}^n \mathbf{X_k}\mathbf{X_k}^T, \qquad \mathbf{S}_2 = \frac{1}{n}\sum_{k=1}^n \mathbf{X_k}^T\mathbf{X_k}. \quad (4)$$

This indicates that the variation on the product graph equals the sum of mode-wise variation, and Cartesian product graph learning encourages signals to be smooth on each factor. In contrast to existing non-consistent GSP methods, we use a log-determinant regularization naturally induced by MLE, which is crucial for the estimator to be consistent as we will show.

GM Interpretation: Consider a random matrix-variate $\boldsymbol{X}$ defined by a IGMRF $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, (\mathbf{L}_1\oplus\mathbf{L}_2)^\dagger)$. Then (3) is the penalized MLE of fitting this model to the product graph signals. Solving (3) amounts to enforcing the Laplacian structure on the Kronecker sum precision matrices. As the Laplacian constraints are essentially a structural prior, they are important for accurate estimation, especially when n is small. We will show that our experiments verify this claim.

3.2 Multi-Way Graph Learning

We now propose the **M**ulti-**W**ay **G**raph (Laplacian) **L**earning (**MWGL**) algorithm for solving (3). First we rewrite (3) as

$$\min_{\mathbf{w}_1,\mathbf{w}_2\geq\mathbf{0}} \Big\{ \mathbf{w}_1^T\mathcal{L}^*\mathbf{S}_1 + \mathbf{w}_2^T\mathcal{L}^*\mathbf{S}_2 - \log\det{}^\dagger(\mathcal{L}\mathbf{w}_1\oplus\mathcal{L}\mathbf{w}_2) + \alpha_1\mathbf{w}_1^T\mathbf{1} + \alpha_2\mathbf{w}_2^T\mathbf{1} \Big\}, \quad (5)$$

since $\mathrm{Tr}(\mathbf{LS}) = \mathrm{Tr}(\mathcal{L}\mathbf{w}\mathbf{S}) = \mathbf{w}^T\mathcal{L}^*\mathbf{S}$. The absolute sign of the ℓ_1 norm of the sparsity regularization is redundant due to the non-negative constraints. We then use projected gradient descent to solve $\mathbf{w}_1$ and $\mathbf{w}_2$. The update of $\mathbf{w}_1$ and $\mathbf{w}_2$ is given by

$$\begin{aligned} \mathbf{w}_1^{(t+1)} &= (\mathbf{w}_1^{(t)} - \eta(\mathcal{L}^*\mathbf{S}_1 - \mathcal{L}^*\mathbf{H}_1^{(t)} + \alpha_1\mathbf{1}))_+, \\ \mathbf{w}_2^{(t+1)} &= (\mathbf{w}_2^{(t)} - \eta(\mathcal{L}^*\mathbf{S}_2 - \mathcal{L}^*\mathbf{H}_2^{(t)} + \alpha_2\mathbf{1}))_+, \end{aligned} \quad (6)$$

where for $\mathbf{H}_1 \in \mathbb{R}^{p_1\times p_1}$ and $\mathbf{H}_2 \in \mathbb{R}^{p_2\times p_2}$ we have

$$\begin{aligned} \mathbf{H}_1 &= \sum_{l=1}^{p_2} (\mathbf{I}_{p_1}\otimes\mathbf{e}_{p_2}^l)^T(\mathcal{L}\mathbf{w}_1\oplus\mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{I}_{p_1}\otimes\mathbf{e}_{p_2}^l), \\ \mathbf{H}_2 &= \sum_{l=1}^{p_1} (\mathbf{e}_{p_1}^l\otimes\mathbf{I}_{p_2})^T(\mathcal{L}\mathbf{w}_1\oplus\mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{e}_{p_1}^l\otimes\mathbf{I}_{p_2}). \end{aligned} \quad (7)$$

The regularization parameters for each factor graph are $\alpha_1 = p_2\alpha$ and $\alpha_2 = p_1\alpha$, but in practice, we can benefit from a free grid search of α_1 and α_2. With the learning rate η of the user's choice, alternating between (6) until stopping criteria is satisfied solves the Cartesian product graph learning problem. The above projected gradient descent scheme is guaranteed to converge in $\mathcal{O}(\frac{1}{t})$ for η that is sufficiently small.

A closer look reveals that this solution is not computationally scalable. Computing the gradient involves taking the inverse of the product graph Laplacian, which can be huge when the number of factors increases. Even for the 2-factor case, the computational cost of inversion will explode quickly as the size of each factor graph grows. Fortunately, we can compute $\mathbf{H}_1$ and $\mathbf{H}_2$ efficiently using the following lemma.

Lemma 1 (Efficient Computation)**.** *The* $\mathbf{H}_1$ *and* $\mathbf{H}_2$ *matrices defined in* (7) *can be efficiently computed as*

$$\begin{aligned}\mathbf{H}_1 &= \mathbf{U}_1 \sum_{l=1}^{p_2} (\boldsymbol{\Lambda}_1 + [\boldsymbol{\Lambda}_2]_{l,l}\mathbf{I}_{p_1})^{\dagger}\mathbf{U}_1^T, \\ \mathbf{H}_2 &= \mathbf{U}_2 \sum_{l=1}^{p_1} (\boldsymbol{\Lambda}_2 + [\boldsymbol{\Lambda}_1]_{l,l}\mathbf{I}_{p_2})^{\dagger}\mathbf{U}_2^T,\end{aligned} \tag{8}$$

where $\mathbf{L}_1 = \mathbf{U}_1\boldsymbol{\Lambda}_1\mathbf{U}_1^T$ *and* $\mathbf{L}_2 = \mathbf{U}_2\boldsymbol{\Lambda}_2\mathbf{U}_2^T$ *are the eigendecompositions of factor Laplacian matrices.*

The key to obtaining (8) is to leverage the spectral structure of the Cartesian product Laplacian and its inverse. A similar strategy has been used in (Yoon and Kim, 2022), but here we do not suffer from the identifiability issue thanks to the Laplacian constraints. See the supplement for the details. Note that (8) only requires matrix operations on the factor scales and avoids taking the cumbersome inversion of $\mathbf{L}$ as in (7). These two steps together demand $\mathcal{O}(p_1^3+p_2^3)$ time complexity (dominated by eigendecompositions), which is a significant reduction from the full matrix inversion $\mathcal{O}(p^3) = \mathcal{O}(p_1^3 p_2^3)$. Alg. 1 summarizes the algorithm.

Algorithm 1 MWGL

```
Input: graph signals $\{\mathbf{X}_k\}$, parameters $\alpha, \eta$
  Compute $\mathbf{S}_1, \mathbf{S}_2$ as in (4).
  Initialize $\mathbf{w}_1$ and $\mathbf{w}_2$.
  repeat
      Compute $\mathbf{H}_1$ and $\mathbf{H}_2$ as in (8).
      Update $\mathbf{w}_1$ and $\mathbf{w}_2$ with (6).
  until convergence or reaching maximum iterations.
Output: factor graph weights $\mathbf{w}_1, \mathbf{w}_2$
```

3.3 Structural Missing Values

Missing values are common in real-world data. In some cases, there are random missing entries in $\mathbf{X}_k$; in other cases, the entire fiber $\{[\mathbf{x}_1]_i, [\mathbf{x}_2]_i, \ldots, [\mathbf{x}_n]_i\}$ of node i is missing. Inferring connectivity of these missing nodes is generally impossible unless the underlying graph is a Cartesian product. An example, which we demonstrate in the experiments, is learning the product graph from multi-view object images when images of some (object, view) pairs are not accessible. For these missing nodes, their object edges are preserved by other views of the same object, and their view edges are preserved by other objects of the same view.

We now propose to learn the graphs and impute the missing values simultaneously. Let $\Psi^{\complement}$ be the set of missing nodes in the product graph. We treat missing values as contamination of the true data and refine the imputation before every projected gradient descent step in Alg. 1, i.e. we alternate between filling in the data and learning the factor graphs as before. Let $\mathbf{X}_k^{(t)}$ be the imputed signals at step t and the signals on the observed nodes $[\mathbf{X}_k^{(t)}]_\Psi = [\mathbf{X}_k]_\Psi$ are fixed. Consider

$$\min_{\{\mathbf{X}_k^{(t)}\}} \Big\{ \frac{1}{2n\beta} \sum_{k=1}^{n} \|\mathbf{X}_k^{(t-1)} - \mathbf{X}_k^{(t)}\|_F^2 + \mathcal{J}(\{\mathbf{X}_k^{(t)}\}) \Big\}, \tag{9}$$

where β is a trade-off parameter. We solve $\{[\mathbf{X}_k^{(t)}]_{\Psi^{\complement}}\}$ inexactly by alternating the following steps

$$\begin{aligned}[\widetilde{\mathbf{X}}_k^{(t-1)}]_{\Psi^{\complement}} &= [(\beta\mathbf{L}_1^{(t-1)} + \mathbf{I}_{p_1})^{-1}\mathbf{X}_k^{(t-1)}]_{\Psi^{\complement}}, \\ [\mathbf{X}_k^{(t)}]_{\Psi^{\complement}} &= [\widetilde{\mathbf{X}}_k^{(t-1)}(\beta\mathbf{L}_2^{(t-1)} + \mathbf{I}_{p_2})^{-1}]_{\Psi^{\complement}}.\end{aligned} \tag{10}$$

(10) is the partial solution of the Tikhonov filtering

$$\begin{aligned}&\min_{\{\widetilde{\mathbf{X}}_k\}} \frac{1}{2n\beta} \sum_{k=1}^{n} \|\mathbf{X}_k^{(t-1)} - \widetilde{\mathbf{X}}_k\|_F^2 + \mathrm{Tr}(\mathbf{L}_1\mathbf{S}_1), \\ &\min_{\{\mathbf{X}_k\}} \frac{1}{2n\beta} \sum_{k=1}^{n} \|\widetilde{\mathbf{X}}_k^{(t-1)} - \mathbf{X}_k\|_F^2 + \mathrm{Tr}(\mathbf{L}_2\mathbf{S}_2),\end{aligned} \tag{11}$$

which are low-pass graph filters that smooth missing value imputations with current factor graph estimation. Note that we alternately filter the signals with factor graphs rather than employ one-pass filtering with the product graph. This eases the computation for the same reason as we stated in Sec. 3.2. We term this algorithm **MWGL-Missing** and summarize it in Alg. 2.

Algorithm 2 MWGL-Missing

```
Input: observed nodes $\Psi, \{[\mathbf{X}_k]_\Psi\}$, parameters $\alpha, \beta, \eta$
  Initialize $\mathbf{w}_1$, $\mathbf{w}_2$.
  repeat
      Refine imputed values $\{[\mathbf{X}_k]_{\Psi^{\complement}}\}$ with (10).
      Update $\mathbf{S}_1, \mathbf{S}_2$ as in (4).
      Compute $\mathbf{H}_1$ and $\mathbf{H}_2$ as in (8).
      Update $\mathbf{w}_1$ and $\mathbf{w}_2$ with (6).
  until convergence or reaching maximum iterations.
Output: factors $\mathbf{w}_1, \mathbf{w}_2$, imputed values$\{[\mathbf{X}_k]_{\Psi^{\complement}}\}$
```

4 THEORETICAL RESULTS

Now we establish the statistical consistency and convergence rates for the Cartesian product Laplacian estimator as in (3). We first make two assumptions regarding the true underlying graph we were to estimate:

(A1) Let $\mathcal{A} = \{(i,j) | [\mathbf{w}]_{i-j+\frac{1}{2}(j-1)(2p-j)} > 0, i > j\}$ be the support set of $\mathbf{w}$. We assume the graph is sparse and the cardinality of the support set is upper bounded by $|\mathcal{A}| \leq s$.

(A2) Let $\{0, \lambda_2, \ldots, \lambda_p\}$ be the eigenvalues of the true product Laplacian in a non-decreasing order. We assume these eigenvalues are bounded away from 0 and ∞ by a constant $z > 1$, such that $\frac{1}{z} \leq \lambda_2 < \lambda_p \leq z$.

Both assumptions are common in high-dimensional statistics literature. Also notice that in our case, bounding the Fiedler value (the second smallest eigenvalue) away from 0 implies that the graph is connected.

Theorem 2 (Existence of MLE)**.** *The penalized negative log-likelihood of Cartesian product Laplacian learning as in* (3) *is lower-bounded, and there exists at least one global minimizer as the solution of the penalized MLE.*

Ying et al. (2021) proved that the negative log-likelihood as in (2) is lower-bounded and the MLE exists. Since the Laplacian of Cartesian product graphs form a subset of all graph Laplacians, the same lower bound applies here. In fact, we derive a tighter lower bound for the Cartesian product graphs. It remains to show that the global minimizer can be achieved in this subspace of Cartesian product graphs. We demonstrate this by parameterizing the product Laplacian in (3) with $\mathbf{w}_1$ and $\mathbf{w}_2$.

Theorem 3 (Uniqueness of MLE)**.** *The objective function of penalized MLE is jointly convex with respect to the factor graphs, and its global minimizer uniquely exists.*

The uniqueness is not surprising since the original graph Laplacian learning problem is convex, and the map from factor graphs to their Cartesian product is linear.

Theorem 4 (High-dimensional consistency)**.** *Suppose assumptions (A1) and (A2) hold. Then with sufficient observations*

$$n \geq \max\left[\frac{c^2 s \log p}{\lambda_p^2 \min(p_1, p_2)}, \frac{c_2^2 \log p}{64 \min(p_1, p_2)}\right], \tag{12}$$

and regularization parameter

$$\alpha = \frac{c_2}{2\lambda_2}\sqrt{\frac{\log p}{n \min(p_1, p_2)}}, \tag{13}$$

the minimizer $\widehat{\mathbf{L}}$ *of the penalized MLE as in* (3) *is asymptotically consistent to the true Laplacian* $\mathbf{L}^*$ *with the Frobenius error bound*

$$\|\widehat{\mathbf{L}} - \mathbf{L}^*\|_F \leq c\sqrt{\frac{\log p}{n \min(p_1, p_2)}}, \tag{14}$$

in probability $1 - 2\exp(-c' \log p)$. c_1, c_2, $c > \frac{8\sqrt{2}c_2\lambda_p^2}{\lambda_2}$, *and* $c' = \frac{c_1 c_2^2}{64} - 2$ *are constants.*

Theorem 4 not only proves that our proposed estimator is guaranteed to converge to the true Laplacian but also improves the rate of consistency from (Ying et al., 2021) by a factor of $\sqrt{\min(p_1, p_2)}$. The improvement reflects the recurrence of factor dependencies in a single product graph signal. A similar trend has been observed in (Greenewald et al., 2019) when the graphical lasso generalizes to multi-dimensional tensors. The key to proving the improved rate is using Hanson-Wright inequality (Hanson and Wright, 1971; Rudelson and Vershynin, 2013) to obtain concentration results on individual modes of the multi-way tensor. For $\min(p_1, p_2) = 1$, our convergence rate coincides with the one in Ying et al. (2021). See detailed proofs of the above theorems in the supplement.

5 EXPERIMENTS

We conduct extensive experiments in MATLAB on both synthetic and real-world datasets to evaluate our method. See the supplement for more details.

5.1 Synthetic Graphs

We first evaluate on synthetic graphs and signals. We use the following models to generate factor graphs:

(1) Erdős-Rényi model with edge probability 0.3;

(2) Barabási-Albert model with preferential attachment 2 starting from 2 initial nodes;

(3) Watts-Strogatz small-world model, where we create an initial ring lattice with degree 2 and rewire every edge of the graph with probability 0.1;

(4) and regular grid model.

Edge weights are then randomly sampled from a uniform distribution $\mathcal{U}(0.1, 2)$ for each edge. We generate weighted factor graphs of $p_1 = 20$ and $p_2 = 25$ nodes using each graph model and take their Cartesian product to obtain graphs of $p = p_1 p_2 = 500$ nodes. The factor grid models are 4×5 grids and 5×5 grids. The signals are then generated from $\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^{\dagger})$. We generate $n = 10 \times 2\hat{}\{0, 1, \ldots, 10\}$ signals for each

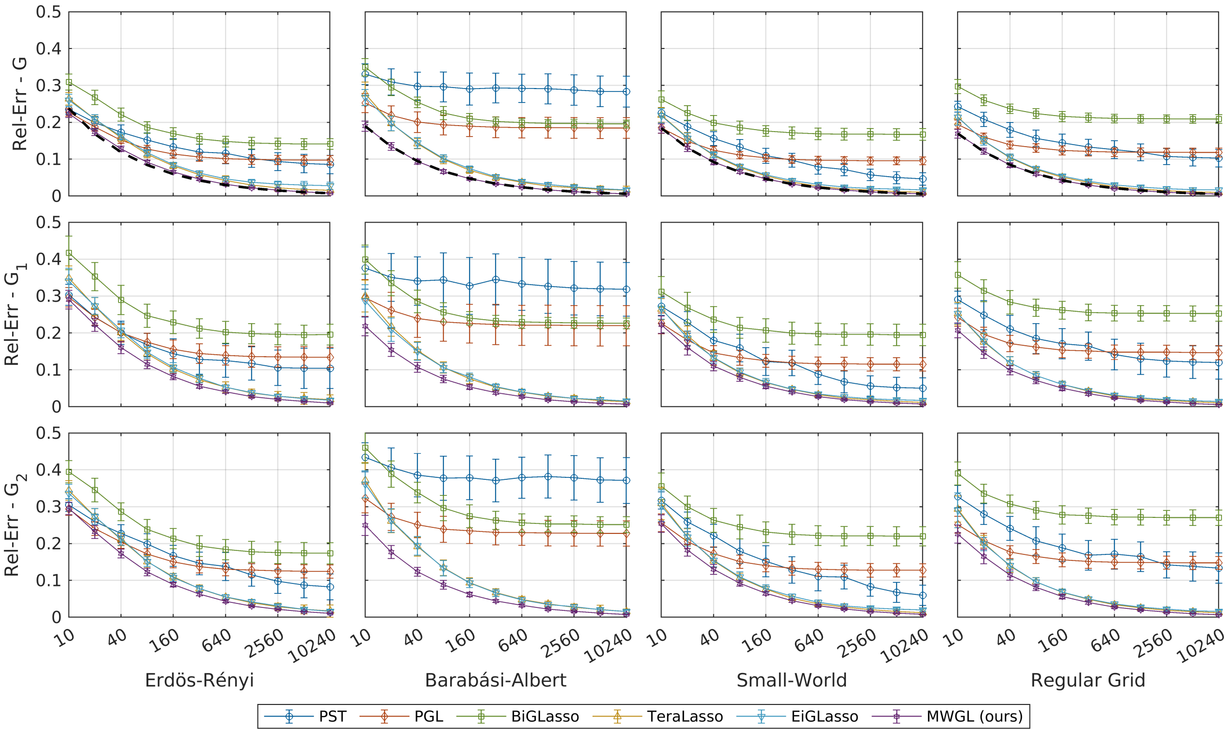


Figure 2: Comparison of different methods on various synthetic data. Each sub-figure shows the trend of Rel-Err of the product or factor Laplacian matrices as n increases. Black dash lines fit the theory in (14) to our results.

synthetic product graph, and evaluate graph learning methods under these different settings. We repeat this process to obtain 50 realizations for each graph model.

We compare MWGL with multiple GSP and GM baselines. For the GSP baselines, we compare with the PGL (Product Graph Learning) method (Kadambari and Chepuri, 2021) and the PST (Product Spectral Template) method (Einizade and Sardouie, 2023). To compare with GM methods, we select the BiGLasso (Kalaitzis et al., 2013), TeraLasso (Greenewald et al., 2019), and EiGLasso (Yoon and Kim, 2022) algorithms that learn precision matrices of Kronecker sum structure. Since a precision matrix $\mathbf{\Theta}$ learned GM methods is generally not a Laplacian, we select its positive "edges" $\mathbf{w_\Theta} = (-\text{tril}(\mathbf{\Theta}, -1))_+$ and build a true Laplacian $\mathcal{L}\mathbf{w_\Theta}$ for evaluation. tril stands for the Matlab operation of lower triangular vectorization.

We use the relative error (Rel-Err) and the area under the precision-recall curve (PR-AUC) as main evaluation metrics. Since the selected GSP baselines impose the constraints $\text{Tr}(\mathbf{L}_1) = p_1$ and $\text{Tr}(\mathbf{L}_2) = p_2$ (thus $\text{Tr}(\mathbf{L}) = \text{Tr}(\mathbf{L_1} \otimes \mathbf{I}_{p_2}) + \text{Tr}(\mathbf{I}_{p_1} \otimes \mathbf{L_2}) = 2p_1p_2$), we normalize the true Laplacian and the Laplacian learned by other methods for the comparison of Rel-Err

$$\mathbf{Ln}_1 = \frac{p_1}{\text{Tr}(\mathbf{L}_1)}\mathbf{L}_1, \mathbf{Ln}_2 = \frac{p_2}{\text{Tr}(\mathbf{L}_2)}\mathbf{L}_2, \mathbf{Ln} = \frac{2p_1p_2}{\text{Tr}(\mathbf{L})}\mathbf{L}.$$

The Rel-Err between the true and learned Laplacian in terms of the Frobenius norm is then computed as

$$\frac{\|\widehat{\mathbf{Ln}} - \mathbf{Ln}^*\|_F}{\|\mathbf{Ln}^*\|_F}, \tag{15}$$

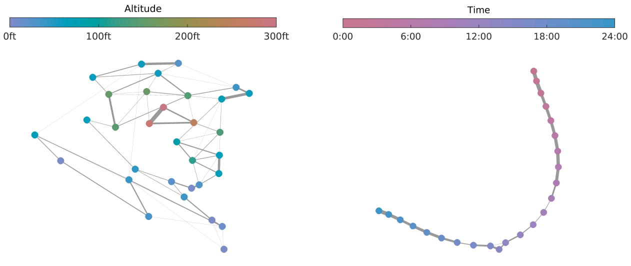
similarly for the factor graphs. We perform grid search to decide the best regularization parameter of each method. Fig. 2 shows the averaged Rel-Err across 50 realizations and the standard deviations on each setting. We leave the PR-AUC results to the supplement.

Fig. 2 demonstrates that our MWGL outperforms both GSP and GM baselines in all the settings. PGL performs well in the low data regime but loses its advantage when n increases. The plots indicate that PGL is inherently asymptotically inconsistent, which is reasonable since their objective function misses the integral log-determinant term of the MLE. PST improves fast as n grows, since the estimated spectral template, i.e. Laplacian eigenvectors, becomes increasingly accurate. However, it still underperforms MWGL even when n is large. For the GM baselines, TeraLasso and EiGLasso outperform BiGLasso with similar performance and come close to MWGL for large n. But when n is small, they underperform MWGL and sometimes other GSP baselines, which shows the importance of the Laplacian constraints as a structural prior. Compared to all these baselines, our MWGL performs well in the full spectrum of n. Also note that the Rel-Err curves of our method fit convergence rate in (14) very well (we solve for c via regression), which validates our theoretical findings.

5.2 Molene Weather Data

We next consider the Molene weather dataset (Loukas and Perraudin, 2019), originally published by the French National Meteorological Service. The dataset contains hourly temperature recordings of 32 weather stations in Brest, France, during the month of January 2014. Our goal is to learn the product of a 32-node geographical graph of weather stations and a 24-node temporal graph of hours. The daily recordings of all stations form a graph signal, and we aim to learn the Cartesian product graph from the 31 daily signals.

MWGL again learns reasonable factors as demonstrated in Fig. 3. The learned weather station graph faithfully reflects their coordinates and altitudes. The 24 nodes of hours form a path graph, in alignment with their temporal order.



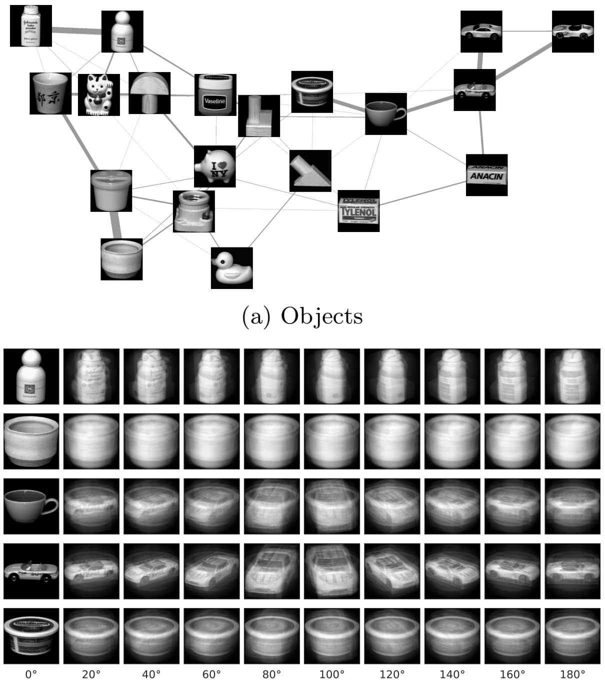
(a) Stations (b) Hours

Figure 3: The inferred factor graphs of Molene. Stations are placed according to their real coordinates.

5.3 COIL-20 Dataset

We now evaluate MWGL-Missing on the Columbia Object Image Library 20 (COIL-20) dataset (Nene et al., 1996). COIL-20 consists of 128×128 grey-scale images of 20 small objects, where each object is placed on a turning table and captured by a fixed camera to obtain multi-views at evenly distributed angles. Images are taken every 5 degrees to produce 72 views per object, which we sub-sample to 36 views. Our goal is to learn a Cartesian product of a 20-node object graph and a 36-node view graph from the $16384 = 128 \times 128$ graph signals. To create structural missingness, we remove the images from 180 to 360 degree of half of the objects (25% of all data) and apply MWGL-Missing.

Fig. 4 shows MWGL-Missing learns meaningful graphs and imputations despite structural missingness. For the object graph, it learns strong connections between the most similar object pairs, such as the car models, and groups other similar objects together. The joint imputation, based on alternating Tikhonov filtering, also reasonably reconstruct the missing images by smoothing the inferred neighboring objects and views. As we can see, imputation of symmetric objects (e.g., last row) rely on the view graph and for imputation of less symmetric objects (e.g., forth row) the object graph plays a more important role. The limitation is that imputing a distinct object (e.g., third row) is generally challenging as it lacks meaningful neighbors.



(a) Objects

(b) Imputed images

Figure 4: (a) The inferred object graph and (b) selected imputations of the COIL-20 dataset. The first column is observed images and other columns are reconstructions across missing angles.

6 CONCLUSIONS

In conclusion, we study the problem of Cartesian product graph learning from multi-way signals. We establish the high-dimensional consistency guarantee for the penalized MLE, which improves the convergence rate on previous general graph Laplacian learning results. We propose an efficient algorithm, MWGL, to solve the penalized MLE, leveraging the Cartesian product structure of the Laplacian. Compared with several GSP and GM baselines, we demonstrate the superiority of our algorithm on both synthetic and real-world datasets. We also provide a joint graph learning and imputation algorithm, MWGL-Missing, and show its efficacy in the presence of structural missingness.

Acknowledgements

GM was partially supported by NSF award CCF-2217058.

References

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516.

Banerjee, O., Ghaoui, L. E., d'Aspremont, A., and Natsoulis, G. (2006). Convex optimization techniques for fitting sparse gaussian graphical models. In *Proceedings of the 23rd international conference on Machine learning*, pages 89–96.

Bassett, D. S. and Sporns, O. (2017). Network neuroscience. *Nature neuroscience*, 20(3):353–364.

Borgatti, S. P., Mehra, A., Brass, D. J., and Labianca, G. (2009). Network analysis in the social sciences. *science*, 323(5916):892–895.

Buciulea, A., Rey, S., and Marques, A. G. (2022). Learning graphs from smooth and graph-stationary signals with hidden variables. *IEEE Transactions on Signal and Information Processing over Networks*, 8:273–287.

Chepuri, S. P., Liu, S., Leus, G., and Hero, A. O. (2017). Learning sparse graphs under smoothness prior. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6508–6512. IEEE.

d'Aspremont, A., Banerjee, O., and El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66.

Dawid, A. P. (1981). Some matrix-variate distribution theory: notational considerations and a bayesian application. *Biometrika*, 68(1):265–274.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, pages 157–175.

Dong, X., Thanou, D., Frossard, P., and Vandergheynst, P. (2016). Learning Laplacian matrix in smooth graph signal representations. *IEEE Trans. Signal Process.*, 64(23):6160–6173.

Dutilleul, P. (1999). The mle algorithm for the matrix normal distribution. *Journal of statistical computation and simulation*, 64(2):105–123.

Egilmez, H. E., Pavez, E., and Ortega, A. (2017). Graph learning from data under Laplacian and structural constraints. *IEEE J. Sel. Topics Signal Process.*, 11(6):825–841.

Einizade, A. and Sardouie, S. H. (2023). Learning product graphs from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Greenewald, K., Zhou, S., and Hero III, A. (2019). Tensor graphical lasso (teralasso). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(5):901–931.

Gupta, A. K. and Nagar, D. K. (1999). *Matrix variate distributions*, volume 104. CRC Press.

Hanson, D. L. and Wright, F. T. (1971). A bound on tail probabilities for quadratic forms in independent random variables. *The Annals of Mathematical Statistics*, 42(3):1079–1083.

He, J., Brugère, T., and Mishne, G. (2023). Product manifold learning with independent coordinate selection. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pages 267–277. PMLR.

Hsieh, C.-J., Dhillon, I., Ravikumar, P., and Sustik, M. (2011). Sparse inverse covariance matrix estimation using quadratic approximation. *Advances in neural information processing systems*, 24.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Kadambari, S. K. and Chepuri, S. P. (2021). Product graph learning from multi-domain data with sparsity and rank constraints. *IEEE Transactions on Signal Processing*, 69:5665–5680.

Kadambari, S. K. and Prabhakar Chepuri, S. (2020). Learning product graphs from multidomain signals. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5665–5669.

Kalaitzis, A., Lafferty, J., Lawrence, N. D., and Zhou, S. (2013). The bigraphical lasso. In *International Conference on Machine Learning*, pages 1229–1237. PMLR.

Kalofolias, V. (2016). How to learn a graph from smooth signals. In *AISTATS*, pages 920–929. PMLR.

Kumar, S., Ying, J., de Miranda Cardoso, J. V., and Palomar, D. P. (2020). A unified framework for structured graph learning via spectral constraints. *JMLR*, 21(22):1–60.

Leng, C. and Tang, C. Y. (2012). Sparse matrix graphical models. *Journal of the American Statistical Association*, 107(499):1187–1200.

Li, L. and Toh, K.-C. (2010). An inexact interior point method for l 1-regularized sparse covariance selection. *Mathematical Programming Computation*, 2:291–315.

Lodhi, M. A. and Bajwa, W. U. (2020). Learning product graphs underlying smooth graph signals. *arXiv preprint arXiv:2002.11277*.

Loukas, A. and Perraudin, N. (2019). Stationary time-vertex signal processing. *EURASIP journal on advances in signal processing*, 2019(1):1–19.

Lu, Z. (2009). Smooth optimization approach for sparse covariance selection. *SIAM Journal on Optimization*, 19(4):1807–1827.

Mateos, G., Segarra, S., Marques, A. G., and Ribeiro, A. (2019). Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Process. Mag.*, 36(3):16–43.

Mazumder, R. and Hastie, T. (2012). The graphical lasso: New insights and alternatives. *Electronic journal of statistics*, 6:2125.

Navarro, M., Wang, Y., Marques, A. G., Uhler, C., and Segarra, S. (2020). Joint inference of multiple graphs from matrix polynomials. *J. Mach. Learn. Res.*, 23:1–35.

Nene, S. A., Nayar, S. K., Murase, H., et al. (1996). Columbia object image library (coil-20).

Ortega, A., Frossard, P., Kovačević, J., Moura, J. M., and Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828.

Oztoprak, F., Nocedal, J., Rennie, S., and Olsen, P. A. (2012). Newton-like methods for sparse inverse covariance estimation. *Advances in neural information processing systems*, 25.

Pasdeloup, B., Gripon, V., Mercier, G., Pastor, D., and Rabbat, M. G. (2017). Characterization and inference of graph diffusion processes from observations of stationary signals. *IEEE Trans. Signal Inf. Process*, 4(3):481–496.

Pavlopoulos, G. A., Secrier, M., Moschopoulos, C. N., Soldatos, T. G., Kossida, S., Aerts, J., Schneider, R., and Bagos, P. G. (2011). Using graph theory to analyze biological networks. *BioData mining*, 4:1–27.

Rudelson, M. and Vershynin, R. (2013). Hanson-wright inequality and sub-gaussian concentration.

Sandryhaila, A. and Moura, J. M. (2014). Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE signal processing magazine*, 31(5):80–90.

Scheinberg, K., Ma, S., and Goldfarb, D. (2010). Sparse inverse covariance selection via alternating linearization methods. *Advances in neural information processing systems*, 23.

Segarra, S., Marques, A. G., Mateos, G., and Ribeiro, A. (2017). Network topology inference from spectral templates. *IEEE Trans. Signal Inf. Process*, 3(3):467–483.

Shafipour, R., Segarra, S., Marques, A. G., and Mateos, G. (2021). Identifying the topology of undirected networks from diffused non-stationary graph signals. *IEEE Open Journal of Signal Processing*, 2:171–189.

Shi, C. and Mishne, G. (2023). Graph laplacian learning with exponential family noise. *arXiv preprint arXiv:2306.08201*.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98.

Slawski, M. and Hein, M. (2015). Estimation of positive definite m-matrices and structure learning for attractive gaussian markov random fields. *Linear Algebra and its Applications*, 473:145–179.

Stanley, J. S., Chi, E. C., and Mishne, G. (2020). Multiway graph signal processing on tensors: Integrative analysis of irregular geometries. *IEEE signal processing magazine*, 37(6):160–173.

Thanou, D., Dong, X., Kressner, D., and Frossard, P. (2017). Learning heat diffusion graphs. *IEEE Trans. Signal Inf. Process*, 3(3):484–499.

Tsiligkaridis, T., Hero III, A. O., and Zhou, S. (2013). On convergence of kronecker graphical lasso algorithms. *IEEE transactions on signal processing*, 61(7):1743–1755.

Wang, Y., Jang, B., and Hero, A. (2020). The sylvester graphical lasso (syglasso). In *International Conference on Artificial Intelligence and Statistics*, pages 1943–1953. PMLR.

Witten, D. M., Friedman, J. H., and Simon, N. (2011). New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Ying, J., de Miranda Cardoso, J. V., and Palomar, D. (2021). Minimax estimation of laplacian constrained precision matrices. In *International Conference on Artificial Intelligence and Statistics*, pages 3736–3744. PMLR.

Yoon, J. H. and Kim, S. (2022). Eiglasso for scalable sparse kronecker-sum inverse covariance estimation. *The Journal of Machine Learning Research*, 23(1):4733–4771.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35.

Zhang, S., Moscovich, A., and Singer, A. (2021). Product manifold learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3241–3249. PMLR.

Zhang, Y. and Schneider, J. (2010). Learning multiple tasks with a sparse matrix-normal penalty. *Advances in neural information processing systems*, 23.

Zhao, L., Wang, Y., Kumar, S., and Palomar, D. P. (2019). Optimization algorithms for graph laplacian estimation via admm and mm. *IEEE Transactions on Signal Processing*, 67(16):4231–4244.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A PROOF OF MAIN THEOREMS AND LEMMAS

A.1 Proof of Lemma 1: Efficient Computation

We first derive $\mathbf{H}_1$ and $\mathbf{H}_2$ as in (7). To obtain $\mathbf{H}_1$, we have

$$
\begin{aligned}
[\nabla_{\mathcal{L}\mathbf{w}_1} \log\det^\dagger(\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)]_{ij} &= \mathrm{Tr}((\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{E}_{p_1}^{ij} \otimes \mathbf{I}_{p_2})) && (16)\\
&= \mathrm{Tr}((\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{E}_{p_1}^{ij} \otimes \sum_{l=1}^{p_2} \mathbf{e}_{p_2}^l {\mathbf{e}_{p_2}^l}^T)) && (17)\\
&= \mathrm{Tr}((\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)^\dagger((\mathbf{I}_{p_1}\mathbf{E}_{p_1}^{ij}\mathbf{I}_{p_1}) \otimes (\sum_{l=1}^{p_2} \mathbf{e}_{p_2}^l 1 {\mathbf{e}_{p_2}^l}^T))) && (18)\\
&= \mathrm{Tr}(\sum_{l=1}^{p_2} (\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)(\mathbf{E}_{p_1}^{ij} \otimes 1)(\mathbf{I}_{p_1} \otimes {\mathbf{e}_{p_2}^l}^T)) && (19)\\
&= \mathrm{Tr}(\sum_{l=1}^{p_2} (\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)^T(\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)\mathbf{E}_{p_1}^{ij}) && (20)\\
&= \left[\sum_{l=1}^{p_2} (\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)^T(\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)\right]_{i,j}, && (21)
\end{aligned}
$$

where $\mathbf{E}_{p_1}^{ij} \in \mathrm{R}^{p_1 \times p_1}$ has one in the $\{i,j\}$-th entry and zeros elsewhere. $\mathbf{H}_2$ can be derived similarly.

We then state the following lemma which characterizes the spectral structure of the Cartesian product Laplacian.

Lemma 5 (Eigen-decomposition of Cartesian Product)**.** *With proper ordering, the eigenvectors of the product graph Laplacian is the Kronecker product of the eigenvectors of the factor graph Laplacian*

$$\mathbf{U} = \mathbf{U}_1 \otimes \mathbf{U}_2,$$

and the eigenvalues of the product graph Laplacian are the Kronecker sum of the eigenvalues of the factor graph Laplacian

$$\mathbf{\Lambda} = \mathbf{\Lambda}_1 \oplus \mathbf{\Lambda}_2.$$

Lemma. 5 follows from the properties of Kronecker product (Barik et al., 2018). Now we proceed to prove Lemma. 1.

Proof. We now derive the efficient computation of $\mathbf{H}_1$ that avoids the expensive large matrix inversion. Let the eigen-decomposition of the factor Laplacians be $\mathcal{L}\mathbf{w}_1 = \mathbf{U}_1\mathbf{\Lambda}_1\mathbf{U}_1^T$ and $\mathcal{L}\mathbf{w}_2 = \mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2^T$. By Lemma 5, we have the eigendecomposition of the product Laplacian

$$\mathbf{L} = \mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2 = (\mathbf{U}_1 \otimes \mathbf{U}_2)(\mathbf{\Lambda}_1 \oplus \mathbf{\Lambda}_2)(\mathbf{U}_1 \otimes \mathbf{U}_2)^T \tag{22}$$

Additionally, we notice that the eigenvectors of $\mathbf{L}^\dagger$ are also $\mathbf{U}_1 \otimes \mathbf{U}_2$. This helps us derive the following

$$
\begin{aligned}
\mathbf{H}_1 &= \sum_{l=1}^{p_2} (\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)^T(\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l) && (23)\\
&= \sum_{l=1}^{p_2} (\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)^T(\mathcal{L}\mathbf{w}_1 \oplus \mathcal{L}\mathbf{w}_2)^\dagger(\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l) && (24)\\
&= \sum_{l=1}^{p_2} (\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)^T(\mathbf{U}_1 \otimes \mathbf{U}_2)(\mathbf{\Lambda}_1 \oplus \mathbf{\Lambda}_2)^\dagger(\mathbf{U}_1 \otimes \mathbf{U}_2)^T(\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l) && (25)\\
&= \sum_{l=1}^{p_2} (\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)^T \sum_{i=1}^{p_1}\sum_{j=1}^{p_2} \frac{1}{[\mathbf{\Lambda_1}]_{i,i} + [\mathbf{\Lambda_2}]_{j,j}}([\mathbf{U}_1]_{:,i} \otimes [\mathbf{U}_2]_{:,j})([\mathbf{U}_1]_{:,i} \otimes [\mathbf{U}_2]_{:,j})^T(\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l) && (26)
\end{aligned}
$$

$$= \sum_{l=1}^{p_2}\sum_{i=1}^{p_1}\sum_{j=1}^{p_2} [\mathbf{U}_2]_{i,j}^2 \frac{1}{[\mathbf{\Lambda_1}]_{i,i} + [\mathbf{\Lambda_2}]_{j,j}} [\mathbf{U}_1]_{:,i}[\mathbf{U}_1]_{:,i}^T \tag{27}$$

$$= \sum_{i=1}^{p_1}\sum_{j=1}^{p_2} \frac{1}{[\mathbf{\Lambda}_1]_{i,i} + [\mathbf{\Lambda}_2]_{j,j}} [\mathbf{U}_1]_{:,i}[\mathbf{U}_1]_{:,i}^T \tag{28}$$

$$= \sum_{j=1}^{p_2} \mathbf{U}_1(\mathbf{\Lambda}_1 + [\mathbf{\Lambda_2}]_{j,j}\mathbf{I}_{p_1})^{\dagger}\mathbf{U}_1^T. \tag{29}$$

Computation of $\mathbf{H}_2$ is derived similarly. □

A.2 Proof of Theorem 2: Existence

Proof. Given $\mathbf{L} = \mathbf{L}_1 \oplus \mathbf{L}_2$, we now prove that the global minimizer of the following penalized MLE

$$\min_{\mathbf{L}\in\Omega_{\mathbf{L}}} \left\{ \mathrm{Tr}(\mathbf{LS}) - \log\det(\mathbf{L}) + \alpha\|\mathbf{L}\|_{1,\text{off}} \right\}, \tag{30}$$

uniquely exists. Provided that both the product and factor graphs are connected, The feasible set over $\mathbf{w}_1$ and $\mathbf{w}_2$ is defined as

$$\Omega_{\mathbf{w}_1,\mathbf{w}_2} := \{(\mathbf{w}_1,\mathbf{w}_2) | \mathbf{w}_1 \geq 0, \mathcal{L}\mathbf{w}_1 + \mathbf{J}_{p_1} \in \mathcal{S}_{++}^{p_2}, \mathbf{w}_2 \geq 0, \mathcal{L}\mathbf{w}_2 + \mathbf{J}_{p_2} \in \mathcal{S}_{++}^{p_2}\}, \tag{31}$$

where $\mathbf{J}_p = \frac{1}{p}\mathbf{1}_p\mathbf{1}_p^T$ and we have $\log\det^{\dagger}(\mathbf{L}) = \log\det(\mathbf{L}+\mathbf{J}_p)$. The conditions $\mathcal{L}\mathbf{w}_1 + \mathbf{J}_{p_1} \in \mathcal{S}_{++}^{p_1}$ and $\mathcal{L}\mathbf{w}_2 + \mathbf{J}_{p_2} \in \mathcal{S}_{++}^{p_2}$ constrain that G_1 and G_2 are connected. Let $\{0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_p\}$ be the eigenvalues of $\mathbf{L}$ and $\mathbf{S}_1$ and $\mathbf{S}_2$ as defined in Section (3.3). We first consider the MLE ($\alpha = 0$) and bound the negative log-likelihood $Q(\mathbf{w}_1, \mathbf{w}_2)$ as below

$$\mathrm{Tr}(\mathbf{LS}) - \log\det(\mathbf{L}) \tag{32}$$

$$= \mathrm{Tr}(\mathbf{LS}) - \log(\prod_{k=2}^{p} \lambda_k) \tag{33}$$

$$\geq \mathrm{Tr}(\mathbf{LS}) - (p-1)\log(\sum_{k=1}^{p}\lambda_k) + (p-1)\log(p-1) \tag{34}$$

$$= \mathrm{Tr}(\mathbf{L}_1\mathbf{S}_1) + \mathrm{Tr}(\mathbf{L}_2\mathbf{S}_2) - (p-1)\log(p_2\sum_{i=1}^{p_1}\lambda_i + p_1\sum_{j=1}^{p_2}\lambda_j) + (p-1)\log(p-1) \tag{35}$$

$$=\langle\mathcal{L}^*(\mathbf{S}_1), \mathbf{w}_1\rangle + \langle\mathcal{L}^*(\mathbf{S}_2), \mathbf{w}_2\rangle - (p-1)\log(p_2\|\mathbf{w}_1\|_1 + p_1\|\mathbf{w}_2\|_1) + (p-1)\log(\frac{p-1}{2}) \tag{36}$$

$$\geq \min(\mathcal{L}^*\tilde{\mathbf{S}}_1 \cup \mathcal{L}^*\tilde{\mathbf{S}}_2)(p_2\|\mathbf{w}_1\|_1 + p_1\|\mathbf{w}_2\|_1) - (p-1)\log(p_2\|\mathbf{w}_1\|_1 + p_1\|\mathbf{w}_2\|_1) + (p-1)\log(\frac{p-1}{2}), \tag{37}$$

where $[\tilde{\mathbf{S}}_1]_{i,j} = \frac{1}{p_2}[\mathbf{S}_1]_{i,j}$ and $[\tilde{\mathbf{S}}_2]_{i,j} = \frac{1}{p_1}[\mathbf{S}_2]_{i,j}$. Inequality (34) holds from the AM-GM inequality, which states that the arithmetic mean of a list of real non-negative numbers is no less than their geometric mean. (35) is attributed to the properties of Cartesian product graphs, and (36) to that the summation of eigenvalues is equal to the trace of the Laplacian. Define the function

$$q(t) = \min(\mathcal{L}^*\tilde{\mathbf{S}}_1 \cup \mathcal{L}^*\tilde{\mathbf{S}}_2)t - (p-1)\log(t) + (p-1)\log(\frac{p-1}{2}). \tag{38}$$

This function is lower-bounded at $t = \frac{p-1}{\min(\mathcal{L}^*\tilde{\mathbf{S}}_1 \cup \mathcal{L}^*\tilde{\mathbf{S}}_2)}$, so long as $\min(\mathcal{L}^*\tilde{\mathbf{S}}_1 \cup \mathcal{L}^*\tilde{\mathbf{S}}_2) > 0$. Therefore, we have that the negative log-likelihood is also lower-bounded

$$Q(\mathbf{w}_1,\mathbf{w}_2) \geq h(p_2\|\mathbf{w}_1\|_1 + p_1\|\mathbf{w}_2\|_1) \geq (p-1)(1 + \log(\frac{\min(\mathcal{L}^*\tilde{\mathbf{S}}_1 \cup \mathcal{L}^*\tilde{\mathbf{S}}_2)}{2})). \tag{39}$$

We then notice that $q(t) \to \infty$ when $t \to \infty$. This is followed with $Q(\mathbf{w}_1, \mathbf{w}_2)$ being coercive, since $\|[\mathbf{w}_1, \mathbf{w}_2]\|_2 \to \infty \rightsquigarrow p_2\|\mathbf{w}_1\|_1 + p_1\|\mathbf{w}_2\|_1 \to \infty$. This indicates that the global minimizer exists in $\mathrm{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2})$.

Furthermore, since the open boundaries $\text{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2})\setminus\Omega_{\mathbf{w}_1,\mathbf{w}_2}$ are results of the connectivity constraint $\mathcal{L}\mathbf{w}_1+\mathbf{J}_{p_1}\succ\mathbf{O}$ and $\mathcal{L}\mathbf{w}_2+\mathbf{J}_{p_2}\succ\mathbf{O}$, we have that $\text{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2})\setminus\Omega_{\mathbf{w}_1,\mathbf{w}_2}$ is a subset of disconnected $\mathbf{w}_1$ and $\mathbf{w}_2$. The set of disconnected $\mathbf{w}_1$ and $\mathbf{w}_2$ is written as

$$\{(\mathbf{w}_1,\mathbf{w}_2)|\det(\mathcal{L}\mathbf{w}_1+\mathbf{J}_{p_1})=0\vee\det(\mathcal{L}\mathbf{w}_2+\mathbf{J}_{p_2})=0\}. \tag{40}$$

Since for the Cartesian product, any factor graph being disconnected leads to the product graph being disconnected, $\forall(\mathbf{w}_1,\mathbf{w}_2)\in\text{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2})\setminus\Omega_{\mathbf{w}_1,\mathbf{w}_2}$ we have $\log\det(\mathbf{L})=-\infty\rightsquigarrow Q(\mathbf{w}_1,\mathbf{w}_2)\to\infty$. This shows that any global minimizer over $\text{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2})$ do not lie on those open boundaries, therefore (30) has at least a global minimizer in $\Omega_{\mathbf{w}_1,\mathbf{w}_2}$ so long as $\min(\mathcal{L}^*\tilde{\mathbf{S}}_1\cup\mathcal{L}^*\tilde{\mathbf{S}}_2)>0$. $\min(\mathcal{L}^*\tilde{\mathbf{S}}_1\cup\mathcal{L}^*\tilde{\mathbf{S}}_2)>0$ holds with probability 1.

For the penalized MLE where $\alpha>0$, we slightly modify (38) to obtain a new lower bound

$$q(t)=(\min(\mathcal{L}^*\tilde{\mathbf{S}}_1\cup\mathcal{L}^*\tilde{\mathbf{S}}_2)+\alpha)t-(p-1)\log(t)+(p-1)\log(\frac{p-1}{2}). \tag{41}$$

As $\min(\mathcal{L}^*\tilde{\mathbf{S}}_1\cup\mathcal{L}^*\tilde{\mathbf{S}}_2)+\alpha>0$ always hold, the penalized MLE always exists. □

A.3 Proof of Theorem 3: Uniqueness

Proof. First we show that $\Omega_{\mathbf{w}_1,\mathbf{w}_2}$ is a convex set. Define the feasible set of $\mathbf{w}_1$ and $\mathbf{w}_2$ as

$$\Omega_{\mathbf{w}_1}:=\{\mathbf{w}_1|\mathbf{w_1}>\mathbf{0},\mathcal{L}\mathbf{w}_1+\mathbf{J}_{p_1}\in\mathcal{S}^{p_1}_{++}\} \tag{42}$$

$$\Omega_{\mathbf{w}_2}:=\{\mathbf{w}_2|\mathbf{w_2}>\mathbf{0},\mathcal{L}\mathbf{w}_2+\mathbf{J}_{p_2}\in\mathcal{S}^{p2}_{++}\}. \tag{43}$$

We can write $\Omega_{\mathbf{w}_1,\mathbf{w}_2}=\Omega_{\mathbf{w}_1}\times\Omega_{\mathbf{w}_2}$. Notice that both $\Omega_{\mathbf{w}_1}$ and $\Omega_{\mathbf{w}_2}$ are convex sets. For any $\mathbf{w}_1^{(0)},\mathbf{w}_1^{(1)}\in\Omega_{\mathbf{w}_1}$ and $\mathbf{w}_2^{(0)},\mathbf{w}_2^{(1)}\in\Omega_{\mathbf{w}_2}$

$$\mathcal{L}\mathbf{w}_1^{(a)}+\mathbf{J}_{p_1}=a(\mathcal{L}\mathbf{w}_1^{(0)}+\mathbf{J}_{p_1})+(1-a)(\mathcal{L}\mathbf{w}_1^{(1)}+\mathbf{J}_{p_1})\in\mathcal{S}^{p_1}_{++},\forall 0<a<1 \tag{44}$$

$$\mathcal{L}\mathbf{w}_2^{(b)}+\mathbf{J}_{p_2}=b(\mathcal{L}\mathbf{w}_2^{(0)}+\mathbf{J}_{p_2})+(1-b)(\mathcal{L}\mathbf{w}_2^{(1)}+\mathbf{J}_{p_2})\in\mathcal{S}^{p2}_{++},\forall 0<b<1, \tag{45}$$

where $\mathbf{w}_1^{(a)}=a\mathbf{w}_1^{(0)}+(1-a)\mathbf{w}_1^{(1)}>\mathbf{0}$ and $\mathbf{w}_2^{(b)}=b\mathbf{w}_2^{(0)}+(1-b)\mathbf{w}_2^{(1)}>\mathbf{0}$, since the PD matrices form a convex cone. Or one can simply realize that the linear interpolation of any two connected graphs (of the same node set) is also connected. Since the direct product of convex sets is a convex set, we know that $\Omega_{\mathbf{w}_1,\mathbf{w}_2}$ is a convex set.

Then, it remains to show that $Q(\mathbf{w}_1,\mathbf{w}_2)$ is a convex function. Define

$$\Omega_{\mathbf{w}}:=\{\mathbf{w}|\mathcal{L}\mathbf{w}\in\Omega_{\mathbf{L}}\}. \tag{46}$$

Since there are bijections between $\Omega_{\mathbf{L}}$, $\Omega_{\mathbf{w}}$, and $\Omega_{\mathbf{w}_1,\mathbf{w}_2}$, from now on, we slightly abuse the notation of the objective function Q and switch back-and-forth upon a suitable parameterization $Q(\mathbf{L})$, $Q(\mathbf{w})$, or $Q(\mathbf{w}_1,\mathbf{w}_2)$. Now we know that $Q(\mathbf{w})$ is a strictly convex function of $\mathbf{w}$, and $\mathbf{w}$ is an affine function of $(\mathbf{w}_1,\mathbf{w}_2)$. This implies that $Q(\mathbf{w}_1,\mathbf{w}_2)$ is also a strictly convex function. Therefore, the global minimizer of Q is unique. □

A.4 Proof of Theorem 4: Consistency

Proof. Now we prove that the penalized MLE in (30) is asymptotically consistent. We use a different proof from (Ying et al., 2021) in spirit that better aligns with the popular route in literature (Rothman et al., 2008; Greenewald et al., 2019). Let $\mathbf{L}^*$ be the Laplacian of the true Cartesian product graph to estimate and $\mathcal{L}\mathbf{w}^*=\mathbf{L}^*$. Let $\mathbf{L}_1^*$ and $\mathbf{L}_2^*$ be the true factor Laplacian, where $\mathcal{L}\mathbf{w}_1^*=\mathbf{L}_1^*$, $\mathcal{L}\mathbf{w}_2^*=\mathbf{L}_2^*$, and $\mathbf{L}^*=\mathbf{L}_1^*\oplus\mathbf{L}_2^*$. Correspondingly, we denote the minimizer of (30) as $\hat{\mathbf{L}}=\hat{\mathbf{L}}_1\oplus\hat{\mathbf{L}}_2$, where $\hat{\mathbf{L}}=\mathcal{L}^*\hat{\mathbf{w}}$, $\hat{\mathbf{L}}_1=\mathcal{L}^*\hat{\mathbf{w}}_1$, and $\hat{\mathbf{L}}_2=\mathcal{L}^*\hat{\mathbf{w}}_2$. We begin with defining a set of perturbations around $\mathbf{L}^*$

$$\mathcal{T}=\{\Delta_{\mathbf{L}}|\Delta_{\mathbf{L}}\in\mathcal{K}_{\mathbf{L}^*},\|\Delta_{\mathbf{L}}\|_F=cr_{n,\mathbf{p}}\}, \tag{47}$$

where $r_{n,\mathbf{p}}=\sqrt{\frac{s\log p}{n\min(p_1,p_2)}}$ for $\mathbf{p}=(p,p_1,p_2)$ and

$$\mathcal{K}_{\mathbf{L}^*}:=\{\Delta_{\mathbf{L}}|\mathbf{L}^*+\Delta_{\mathbf{L}}\in\Omega_{\mathbf{L}}\}. \tag{48}$$

Define the following convex function over $\mathcal{T}$

$$F(\Delta_{\mathbf{L}}) = Q(\mathbf{L}^* + \Delta_{\mathbf{L}}) - Q(\mathbf{L}^*). \tag{49}$$

Our goal now is to show that

$$F(\Delta_{\mathbf{L}}) > 0, \forall \Delta_{\mathbf{L}} \in \mathcal{T}. \tag{50}$$

To see the rationale behind (50), notice that

$$F(\hat{\mathbf{L}} - \mathbf{L}^*) = Q(\hat{\mathbf{L}}) - Q(\mathbf{L}^*) \leq 0, \tag{51}$$

since $\hat{\mathbf{L}}$ minimize $Q(\mathbf{L})$. Provided that $F(\Delta_{\mathbf{L}})$ is a convex function, (50) ultimately implies that

$$\|\hat{\mathbf{L}} - \mathbf{L}^*\|_F \leq c r_{n,\mathbf{p}}. \tag{52}$$

To prove this is true, we first prove the following

$$F(\Delta_{\mathbf{L}}) > 0, \forall \Delta_{\mathbf{L}} \in \mathcal{K}_{\mathbf{L}^*}, \|\Delta_{\mathbf{L}}\|_F > c r_{n,\mathbf{p}}. \tag{53}$$

By contradiction, suppose there exists a $\Delta'_{\mathbf{L}} \in \mathcal{K}_{\mathbf{L}^*}$, such that $\|\Delta'_{\mathbf{L}}\|_F > c r_{n,\mathbf{p}}$ and $F(\Delta'_{\mathbf{L}}) < 0$. Let $\theta = \frac{c r_{n,\mathbf{p}}}{\|\Delta'_{\mathbf{L}}\|_F} < 1$. Then

$$F(\theta\Delta'_{\mathbf{L}}) = F((1-\theta)\mathbf{O} + \theta\Delta'_{\mathbf{L}}) \leq (1-\theta)F(\mathbf{O}) + \theta F(\Delta'_{\mathbf{L}}) = \theta F(\Delta'_{\mathbf{L}}) < 0. \tag{54}$$

This contradicts with (50) since $\theta\Delta'_{\mathbf{L}} \in \mathcal{T}$. Thus (52) must holds under (50).

Now we move forward to prove (50). We write out $F(\Delta_{\mathbf{L}})$

$$F(\Delta_{\mathbf{L}}) = \mathrm{Tr}(\Delta_{\mathbf{L}}\mathbf{S}) - (\log\det(\mathbf{L}^* + \Delta_{\mathbf{L}} + \mathbf{J}_p) - \log\det(\mathbf{L}^* + \mathbf{J}_p)) + \alpha(\|\mathbf{L}^* + \Delta_{\mathbf{L}}\|_{1,\text{off}} - \|\mathbf{L}^*\|_{1,\text{off}}). \tag{55}$$

Consider the Taylor's expansion of $\log\det(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)$ with the integral remainder

$$\log\det(\mathbf{L}^* + \Delta_{\mathbf{L}} + \mathbf{J}_p) - \log\det(\mathbf{L}^* + \mathbf{J}_p) = \mathrm{Tr}((\mathbf{L}^* + \mathbf{J}_p)^{-1}\Delta_{\mathbf{L}}) + \int_0^1 (1-\nu)\nabla^2_\nu \log\det(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p) d\nu, \tag{56}$$

and further the remainder

$$\int_0^1 (1-\nu)\nabla^2_\nu \log\det(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p) d\nu = -\mathrm{vec}(\Delta_{\mathbf{L}})^T (\int_0^1 (1-\nu)(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} \otimes (\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} d\nu) \mathrm{vec}(\Delta_{\mathbf{L}}). \tag{57}$$

Therefore we have

$$F(\Delta_{\mathbf{L}}) = I_1 + I_2 + I_3, \tag{58}$$

where

$$I_1 = \mathrm{Tr}(\Delta_{\mathbf{L}}(\mathbf{S} - (\mathbf{L}^* + \mathbf{J}_p)^{-1})), \tag{59}$$

$$I_2 = \mathrm{vec}(\Delta_{\mathbf{L}})^T (\int_0^1 (1-\nu)(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} \otimes (\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} d\nu) \mathrm{vec}(\Delta_{\mathbf{L}}), \tag{60}$$

$$I_3 = \alpha(\|\mathbf{L}^* + \Delta_{\mathbf{L}}\|_{1,\text{off}} - \|\mathbf{L}^*\|_{1,\text{off}}). \tag{61}$$

We now bound each term separately.

Bound I_1: We follow the argument in Ying et al. (2020) and assume that the graph signals are sampled from the process referred to as conditioning by Kriging (Rue and Held, 2005). This process first sample from the proper GMRF $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, (\mathbf{L}^* + \mathbf{J}_p)^{-1})$, then correct these samples by subtracting their mean to make them DC-intrinsic

$$\bar{\mathbf{x}} = \mathbf{x} - \frac{1}{p}\mathbf{1}\mathbf{1}^T\mathbf{x}. \tag{62}$$

Let $\mathbf{\Sigma} = (\mathbf{L}^* + \mathbf{J}_p)^{-1}$ be the covariance matrix of the original proper GMRF. Since $\Delta_{\mathbf{L}} \in \mathcal{K}_{\mathbf{L}^*}$, we have

$$\begin{aligned} I_1 &= \mathrm{Tr}((\Delta_{\mathbf{L}_1} \oplus \Delta_{\mathbf{L}_2})(\mathbf{S} - (\mathbf{L}^* + \mathbf{J}_p)^{-1})) && (63)\\ &= \mathrm{Tr}(\Delta_{\mathbf{L}_1}(\mathbf{S}_1 - \mathbf{\Sigma}_1)) + \mathrm{Tr}(\Delta_{\mathbf{L}_2}(\mathbf{S}_2 - \mathbf{\Sigma}_2)) && (64)\\ &= \Delta_{\mathbf{w}_1}^T \mathcal{L}^*(\mathbf{S}_1 - \mathbf{\Sigma}_1) + \Delta_{\mathbf{w}_2}^T \mathcal{L}^*(\mathbf{S}_2 - \mathbf{\Sigma}_2) && (65)\\ &= p_2 \Delta_{\mathbf{w}_1}^T \mathcal{L}^*(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1) + p_1 \Delta_{\mathbf{w}_2}^T \mathcal{L}^*(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2). && (66) \end{aligned}$$

where $\Delta_{\mathbf{L}} = \Delta_{\mathbf{L}_1} \oplus \Delta_{\mathbf{L}_2}$ and $[\tilde{\mathbf{\Sigma}}_1]_{i,j} = \frac{1}{p_2}[\mathbf{\Sigma}_1]_{i,j}$ and $[\tilde{\mathbf{\Sigma}}_2]_{i,j} = \frac{1}{p_1}[\mathbf{\Sigma}_2]_{i,j}$. $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ are defined similarly as in the Section 3.2.

$$\begin{aligned} \mathbf{\Sigma}_1 &= \sum_{l=1}^{p_2} (\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l)^T \mathbf{\Sigma} (\mathbf{I}_{p_1} \otimes \mathbf{e}_{p_2}^l), && (67)\\ \mathbf{\Sigma}_2 &= \sum_{l=1}^{p_1} (\mathbf{e}_{p_1}^l \otimes \mathbf{I}_{p_2})^T \mathbf{\Sigma} (\mathbf{e}_{p_1}^l \otimes \mathbf{I}_{p_2}). && (68) \end{aligned}$$

We then have

$$\begin{aligned} [\mathcal{L}^* \mathbf{S}_1]_{i-j+\frac{1}{2}(j-1)(2p_1-j)} &= \frac{1}{n} \sum_{k=1}^{n} \sum_{l=1}^{p_2} ([\mathbf{x}_k]_{(i-1)p_2+l} - [\mathbf{x}_k]_{(j-1)p_2+l})^2, \forall 1 \leq j < i \leq p_1, && (69)\\ [\mathcal{L}^* \mathbf{S}_2]_{i-j+\frac{1}{2}(j-1)(2p_2-j)} &= \frac{1}{n} \sum_{k=1}^{n} \sum_{l=1}^{p_1} ([\mathbf{x}_k]_{(l-1)p_2+i} - [\mathbf{x}_k]_{(l-1)p_2+j})^2, \forall 1 \leq j < i \leq p_2. && (70) \end{aligned}$$

Here we focus on $\mathcal{L}^* \mathbf{S}_1$, and results on $\mathcal{L}^* \mathbf{S}_2$ can be derived similarly. Let $m_1 = i - j + \frac{1}{2}(j-1)(2p_1 - j)$. We rewrite $[\mathcal{L}^* \tilde{\mathbf{S}}_1]_{m_1}$ into the quadratic form of the entries of $\mathbf{x}$

$$[\mathcal{L}^* \mathbf{S}_1]_{m_1} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k^T (\mathcal{L}\mathbf{e}_m \otimes \mathbf{I}_{p_2}) \mathbf{x}_k, \forall 1 \leq j < i \leq p_1, \tag{71}$$

where $\mathbf{e}_{m_1} \in \mathbb{R}^{\frac{p_1(p_1-1)}{2}}$ has one in the $\{i - j + \frac{1}{2}(j-1)(2p_1 - j)\}$-th entry and zeros otherwise. Let $\mathbf{x}_k = \mathbf{\Sigma}^{\frac{1}{2}} \mathbf{z}_k$, where $\boldsymbol{z_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$ is the source signal of the GSP system. We then write the above quadratic as

$$[\mathcal{L}^* \mathbf{S}_1]_{m_1} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{z}_k^T \mathbf{\Sigma}^{\frac{1}{2}} (\mathcal{L}\mathbf{e}_{m_1} \otimes \mathbf{I}_{p_2}) \mathbf{\Sigma}^{\frac{1}{2}} \mathbf{z}_k, \forall 1 \leq j < i \leq p_1, \tag{72}$$

Let $\mathbf{M}_{i,j} = \mathbf{\Sigma}^{\frac{1}{2}} (\mathcal{L}\mathbf{e}_{m_1} \otimes \mathbf{I}_{p_2}) \mathbf{\Sigma}^{\frac{1}{2}}$. By the Hanson-Wright inequality Hanson and Wright (1971); Rudelson and Vershynin (2013), we have

$$\begin{aligned} \mathbb{P}\left\{ |\frac{1}{n} \sum_{k=1}^{n} \mathbf{z}_k^T \mathbf{M}_{i,j} \mathbf{z}_k - \mathbb{E}[\frac{1}{n} \sum_{k=1}^{n} \mathbf{z}_k^T \mathbf{M}_{i,j} \mathbf{z}_k]| > h \right\} &\leq 2\exp\left[-c_1 \min\left(\frac{nh^2}{K^4 \|\mathbf{M}_{i,j}\|_F^2}, \frac{nh}{K^2 \|\mathbf{M}_{i,j}\|_2} \right) \right] && (73)\\ &\leq 2\exp\left[-c_1 \min\left(\frac{nh^2}{4K^4 p_2 \|\mathbf{\Sigma}\|_2^2}, \frac{nh}{2K^2 \|\mathbf{\Sigma}\|_2} \right) \right] && (74)\\ &\leq 2\exp\left[-c_1 \min\left(\frac{nh^2}{64 p_2 \|\mathbf{\Sigma}\|_2^2}, \frac{nh}{8 \|\mathbf{\Sigma}\|_2} \right) \right], && (75) \end{aligned}$$

where $K = 2$ is the sub-Gaussian norm of $\mathbf{z}_k$. (74) holds by the properties of matrix norms and the trace inequalities (Fang et al., 1994)

$$\begin{aligned} \|\mathbf{M}_{i,j}\|_F^2 &\leq \|\mathbf{\Sigma}^{\frac{1}{2}}\|_2^2 \|\mathcal{L}\mathbf{e}_{m_1} \otimes \mathbf{I}_{p_2})\|_F^2 \|\mathbf{\Sigma}^{\frac{1}{2}}\|_2^2 = 4p_2 \|\mathbf{\Sigma}\|_2^2, && (76)\\ \|\mathbf{M}_{i,j}\|_2 &\leq \|\mathbf{\Sigma}^{\frac{1}{2}}\|_2 \|\mathcal{L}\mathbf{e}_{m_1} \otimes \mathbf{I}_{p_2})\|_2 \|\mathbf{\Sigma}^{\frac{1}{2}}\|_2 = 2\|\mathbf{\Sigma}\|_2, && (77) \end{aligned}$$

where $\|\mathcal{L}\mathbf{e}_{m_1} \otimes \mathbf{I}_{p_2})\|_F^2 = 4p_2$ and $\|\mathcal{L}\mathbf{e}_{m_1} \otimes \mathbf{I}_{p_2})\|_2 = \|\mathcal{L}\mathbf{e}_{m_1}\|_2 = 2$. Let $\epsilon = \frac{h}{\sqrt{p_2}\|\mathbf{\Sigma}\|_2}$ and plug (72) into (75)

$$\mathbb{P}\left\{|[\mathcal{L}^*\mathbf{S}_1]_{m_1} - \mathbb{E}[[\mathcal{L}^*\mathbf{S}_1]_{m_1}]| > \epsilon\sqrt{p_2}\|\mathbf{\Sigma}\|_2\right\} \leq 2\exp(-\frac{cn\epsilon^2}{64}), \forall\epsilon \leq 8\sqrt{p_2}. \tag{78}$$

Meanwhile

$$\begin{aligned}
\mathbb{E}[[\mathcal{L}^*\mathbf{S}_1]_{m_1}] &= \frac{1}{n}\sum_{k=1}^{n}\sum_{l=1}^{p_2}\mathbb{E}[([\mathbf{x}_k]_{(i-1)p_2+l} - [\mathbf{x}_k]_{(j-1)p_2+l})^2] && (79)\\
&= \frac{1}{n}\sum_{k=1}^{n}\sum_{l=1}^{p_2}\mathbb{E}[[\mathbf{x}_k]^2_{(i-1)p_2+l}] - 2\mathbb{E}[[\mathbf{x}_k]_{(i-1)p_2+l}[\mathbf{x}_k]_{(j-1)p_2+l}] + \mathbb{E}[[\mathbf{x}_k]^2_{(j-1)p_2+l}] && (80)\\
&= [\mathbf{\Sigma}_1]_{i,i} - [\mathbf{\Sigma}_1]_{i,j} - [\mathbf{\Sigma}_1]_{j,i} + [\mathbf{\Sigma}_1]_{j,j} && (81)\\
&= [\mathcal{L}^*\mathbf{\Sigma}_1]_{m_1}. && (82)
\end{aligned}$$

Therefore

$$\mathbb{P}\left\{|[\mathcal{L}^*(\mathbf{S}_1 - \mathbf{\Sigma}_1)]_{m_1}| > \epsilon\sqrt{p_2}\|\mathbf{\Sigma}\|_2\right\} \leq 2\exp(-\frac{c_1n\epsilon^2}{64}), \forall\epsilon \leq 8\sqrt{p_2}, \tag{83}$$

and we reach the following concentration result for $\mathcal{L}^*\tilde{\mathbf{S}}_1$

$$\mathbb{P}\left\{|[\mathcal{L}^*(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1)]_{m_1}| > \frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{p_2}}\right\} \leq 2\exp(-\frac{c_1n\epsilon^2}{64}), \forall\epsilon \leq 8\sqrt{p_2}, \tag{84}$$

Similarly for $\mathcal{L}^*\tilde{\mathbf{S}}_2$ we derive for $m_2 = i - j + \frac{1}{2}(j-1)(2p_2 - j)$

$$\mathbb{P}\left\{|[\mathcal{L}^*(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2)]_{m_2}| > \frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{p_1}}\right\} \leq 2\exp(-\frac{c_1n\epsilon^2}{64}), \forall\epsilon \leq 8\sqrt{p_1}. \tag{85}$$

By the union bound

$$\begin{aligned}
&\mathbb{P}\left\{\max\left[|\mathcal{L}^*(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1)|, |\mathcal{L}^*(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2)|\right] > \frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{\min(p_1, p_2)}}\right\} && (86)\\
\leq&\mathbb{P}\left\{\max_{m_1}|[\mathcal{L}^*(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1)]_{m_1}| > \frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{p_2}}\right\} + \mathbb{P}\left\{\max_{m_2}|[\mathcal{L}^*(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2)]_{m_2}| > \frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{p_1}}\right\} && (87)\\
\leq&\sum_{m_1=1}^{\frac{p_1(p_1-1)}{2}} 2\exp(-\frac{c_1n\epsilon^2}{64}) + \sum_{m_2=1}^{\frac{p_2(p_2-1)}{2}} 2\exp(-\frac{c_1n\epsilon^2}{64}) && (88)\\
\leq&2\max^2(p_1, p_2)\exp(-\frac{c_1n\epsilon^2}{64}). && (89)
\end{aligned}$$

By calculation we then have

$$\mathbb{P}\left\{\max\left[|\mathcal{L}^*(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1)|, |\mathcal{L}^*(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2)|\right] \leq \frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{\min(p_1, p_2)}}\right\} \geq 1 - 2\max^2(p_1, p_2)\exp(-\frac{c_1n\epsilon^2}{64}). \tag{90}$$

So with the probability stated in (90), we derive the following lower bound for I_1

$$\begin{aligned}
I_1 &= p_2\operatorname{Tr}(\Delta_{\mathbf{L}_1}(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1)) + p_1\operatorname{Tr}(\Delta_{\mathbf{L}_2}(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2)) && (91)\\
&\geq -p_2|\Delta_{\mathbf{w}_1}^T\mathcal{L}^*(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1)| - p_1|\Delta_{\mathbf{w}_2}^T\mathcal{L}^*(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2)| && (92)\\
&\geq -\max\left[|\mathcal{L}^*(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1)|, |\mathcal{L}^*(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2)|\right](p_2|\Delta_{\mathbf{w}_1}| + p_1|\Delta_{\mathbf{w}_2}|) && (93)\\
&\geq -\frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{\min(p_1, p_2)}}|\Delta_{\mathbf{w}}|. && (94)
\end{aligned}$$

Bound I_2: From the min-max theorem, we have

$$I_2 \geq \|\Delta_{\mathbf{L}}\|_F^2 \lambda_{\min}(\int_0^1 (1-\nu)(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} \otimes (\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} d\nu). \tag{95}$$

Then given the convexity of $\lambda_{\max}(\cdot)$ and concavity of $\lambda_{\min}(\cdot)$

$$\lambda_{\min}(\int_0^1 (1-\nu)(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} \otimes (\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} d\nu) \tag{96}$$

$$\geq \int_0^1 (1-\nu)\lambda_{\min}^2(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} d\nu \tag{97}$$

$$\geq \min_{\nu\in[0,1]} \left[\lambda_{\min}^2(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1}\right] \int_0^1 (1-\nu) d\nu \tag{98}$$

$$= \frac{1}{2} \min_{\nu\in[0,1]} \left[\frac{1}{\lambda_{\max}^2(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)}\right] \tag{99}$$

$$= \frac{1}{2\max_{\nu\in[0,1]}\left[\lambda_{\max}^2(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)\right]} \tag{100}$$

$$\geq \frac{1}{2\max_{\nu\in[0,1]}^2\left[(\lambda_{\max}(\mathbf{L}^* + \mathbf{J}_p) + \|\nu\Delta_{\mathbf{L}}\|_2)\right]} \tag{101}$$

$$= \frac{1}{2(\lambda_{\max}(\mathbf{L}^* + \mathbf{J}_p) + \|\Delta_{\mathbf{L}}\|_2)^2} \tag{102}$$

Then with n sufficiently large

$$n \geq \frac{c^2 s \log p}{\lambda_{\max}^2(\mathbf{L}^* + \mathbf{J}_p)\min(p_1, p_2)}, \tag{103}$$

such that $\|\Delta_{\mathbf{L}}\|_2 \leq \|\Delta_{\mathbf{L}}\|_F \leq \lambda_{\max}(\mathbf{L}^* + \mathbf{J}_p)$, we obtain a lower bound for I_2

$$I_2 \geq \frac{\|\Delta_{\mathbf{L}}\|_F^2}{2(\lambda_{\max}(\mathbf{L}^* + \mathbf{J}_p) + \|\Delta_{\mathbf{L}}\|_2)^2} \tag{104}$$

$$\geq \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\lambda_{\max}^2(\mathbf{L}^* + \mathbf{J}_p)}. \tag{105}$$

Bound I_3: To bound I_3, we use triangular inequality

$$\|\mathbf{L}^* + \Delta_{\mathbf{L}}\|_{1,\text{off}} - \|\mathbf{L}^*\|_{1,\text{off}} = \|\mathbf{L}^* + \Delta_{\mathbf{L}}\|_{1,\mathcal{A}} + \|\Delta_{\mathbf{L}}\|_{1,\mathcal{A}^{\complement}} - \|\mathbf{L}^*\|_{1,\mathcal{A}} \geq \|\Delta_{\mathbf{L}}\|_{1,\mathcal{A}^{\complement}} - \|\Delta_{\mathbf{L}}\|_{1,\mathcal{A}}, \tag{106}$$

to obtain

$$I_3 \geq \alpha(\|\Delta_{\mathbf{L}}\|_{1,\mathcal{A}^{\complement}} - \|\Delta_{\mathbf{L}}\|_{1,\mathcal{A}}) = 2\alpha|\Delta_{\mathbf{w}}|_{1,\mathcal{A}^{\complement}} - 2\alpha|\Delta_{\mathbf{w}}|_{1,\mathcal{A}}. \tag{107}$$

Bound $I_1 + I_2 + I_3$: So overall

$$F(\Delta_{\mathbf{L}}) \geq -\frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{\min(p_1, p_2)}}|\Delta_{\mathbf{w}}|_1 + \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\lambda_{\max}^2(\mathbf{L}^* + \mathbf{J}_p)} + 2\alpha|\Delta_{\mathbf{w}}|_{1,\mathcal{A}^{\complement}} - 2\alpha|\Delta_{\mathbf{w}}|_{1,\mathcal{A}} \tag{108}$$

$$= \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\lambda_{\max}^2(\mathbf{L}^* + \mathbf{J}_p)} - (\frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{\min(p_1, p_2)}} - 2\alpha)|\Delta_{\mathbf{w}}|_{1,\mathcal{A}^{\complement}} - (\frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{\min(p_1, p_2)}} + 2\alpha)|\Delta_{\mathbf{w}}|_{1,\mathcal{A}}. \tag{109}$$

Let $\epsilon = c_2\sqrt{\frac{\log p}{n}}$ with sufficiently large

$$n \geq \frac{c_2^2 \log p}{64\min(p_1, p_2)}, \tag{110}$$

so that $\epsilon \leq 8\sqrt{\min(p_1, p_2)}$. Then choose

$$\alpha \geq \frac{c_2\|\mathbf{\Sigma}\|_2}{2}\sqrt{\frac{\log p}{n\min(p_1, p_2)}}, \tag{111}$$

such that $\epsilon\|\mathbf{\Sigma}\|_2 - 2\alpha\sqrt{\min(p_1,p_2)} \leq 0$. Also note that

$$|\Delta_{\mathbf{w}}|_{1,\mathcal{A}} \leq \sqrt{s}\|\Delta_{\mathbf{w}}\|_{2,\mathcal{A}} \leq \sqrt{s}\|\Delta_{\mathbf{w}}\|_2 \leq \sqrt{\frac{s}{2}}\|\Delta_{\mathbf{L}}\|_F, \tag{112}$$

then with $\frac{\alpha}{\gamma} = \frac{c_2\|\mathbf{\Sigma}\|_2}{2}\sqrt{\frac{\log p}{n\min(p_1,p_2)}}$ we obtain

$$\begin{aligned}
F(\Delta_{\mathbf{L}}) &\geq \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\lambda_{\max}^2(\mathbf{L}^*+\mathbf{J}_p)} - (1+\gamma)c_2\|\mathbf{\Sigma}\|_2\sqrt{\frac{\log p}{n\min(p_1,p_2)}}|\Delta_{\mathbf{w}}|_{1,\mathcal{A}} && (113)\\
&\geq \|\Delta_{\mathbf{L}}\|_F^2\Big(\frac{1}{8\lambda_{\max}^2(\mathbf{L}^*+\mathbf{J}_p)} - (1+\gamma)c_2\|\mathbf{\Sigma}\|_2\sqrt{\frac{s\log p}{2n\min(p_1,p_2)}}\|\Delta_{\mathbf{L}}\|_F^{-1}\Big) && (114)\\
&= \|\Delta_{\mathbf{L}}\|_F^2\Big(\frac{1}{8\lambda_{\max}^2(\mathbf{L}^*+\mathbf{J}_p)} - (1+\gamma)\frac{c_2\|\mathbf{\Sigma}\|_2}{\sqrt{2}c}\Big) && (115)\\
&> 0, && (116)
\end{aligned}$$

so long as c is sufficiently large

$$c > 4\sqrt{2}(1+\gamma)c_2\|\mathbf{\Sigma}\|_2\lambda_{\max}^2(\mathbf{L}^*+\mathbf{J}_p). \tag{117}$$

This holds with probability at least

$$\begin{aligned}
&\mathbb{P}\left\{\max\Big[|\mathcal{L}^*(\tilde{\mathbf{S}}_1 - \tilde{\mathbf{\Sigma}}_1)|, |\mathcal{L}^*(\tilde{\mathbf{S}}_2 - \tilde{\mathbf{\Sigma}}_2)|\Big] \leq \frac{\epsilon\|\mathbf{\Sigma}\|_2}{\sqrt{\min(p_1,p_2)}}\right\} && (118)\\
&\geq 1 - 2\max{}^2(p_1,p_2)\exp\big(-\frac{c_1 n\epsilon^2}{64}\big) && (119)\\
&= 1 - 2\exp\Big[2\log\left[\max(p_1,p_2)\right] - \frac{c_1c_2^2}{64}\log p\Big] && (120)\\
&\geq 1 - 2\exp\left(-c'\log p\right), && (121)
\end{aligned}$$

where $c' = \frac{c_1c_2^2}{64} - 2$. $\gamma \geq 1$ here is a tuning parameter. Setting $\gamma = 1$ retrieves Theorem. 4. □

B EXPERIMENTAL DETAILS

We now detail our experimental settings. To initialize $\mathbf{w}_1$ and $\mathbf{w}_2$, we first calculate $\mathbf{S}_1^{-1}$ and $\mathbf{S}_2^{-1}$ to obtain an initial guess of the factor precision matrices. The non-Laplacian matrices are then processed as described in Sec. 5.1 to initialize $\mathbf{w}_1 = (-\mathrm{tril}(\mathbf{S}_1^{-1}, -1))_+$ and $\mathbf{w}_2 = (-\mathrm{tril}(\mathbf{S}_2^{-1}, -1))_+$. When there are missing values, we use the largest sub-columns/rows of $\mathbf{X}_k$ that do not include missing entries to compute

$$\mathbf{S}_1 = \frac{1}{n}\sum_{k=1}^{n}[\mathbf{X}_k]_{:,\Psi_c}[\mathbf{X}_k]_{:,\Psi_c}{}^T, \qquad \mathbf{S}_2 = \frac{1}{n}\sum_{k=1}^{n}[\mathbf{X}_k]_{\Psi_r,:}{}^T[\mathbf{X}_k]_{\Psi_r,:}. \tag{122}$$

Here Ψ_c and Ψ_r are defined as the sets of all the columns and rows that do not contain missing values. For the initial imputation of a node $(r_1,r_2) \in \Psi^{\complement}$, we consider the set of node pairs $\Psi_{(r_1,r_2)} = \{(r_1,i_2) \notin \Psi^{\complement}\} \vee \{(i_1,r_2) \notin \Psi^{\complement}\}$, and use

$$[\mathbf{X}_k]_{r_1,r_2} = \frac{1}{|\Psi_{(r_1,r_2)}|}\sum_{(i_1,i_2)\in\Psi_{(r_1,r_2)}}[\mathbf{X}_k]_{i_1,i_2}, \tag{123}$$

i.e. average of all non-missing entries that belong to the same column or row with itself. For all experiments, we set the learning rate η of project gradient descent to 1e-3 and the tolerance ϵ to be 1e-6. For pre-processing steps, we normalize the COIL-20 images with $\frac{\mathbf{X}}{255} - 0.5$ and remove the station and hour means of the Molene data.

We use official implementations for the PGL, BiGLasso, and TeraLasso baselines. For the PST baseline, since the original paper proposed to learn the eigenvectors of factor adjacency matrices (Einizade and Sardouie, 2023), we implement an adapted algorithm that learns eigenvectors of factor Laplacian matrices. Such adaptation is presented in the Sec. 5C of (Segarra et al., 2017).

B.1 PR-AUC of Edge Estimation of Synthetic Graphs

To compute PR-AUC, we use $\mathbb{1}_{\mathbf{w}>\rho}(\hat{\mathbf{w}})$ as the binary edge predictions of an increasing series of threshold ρ and calculate area of the precision-recall curves. The results are shown in Fig. 5. We can see that MWGL again outperforms all the baselines on different settings.

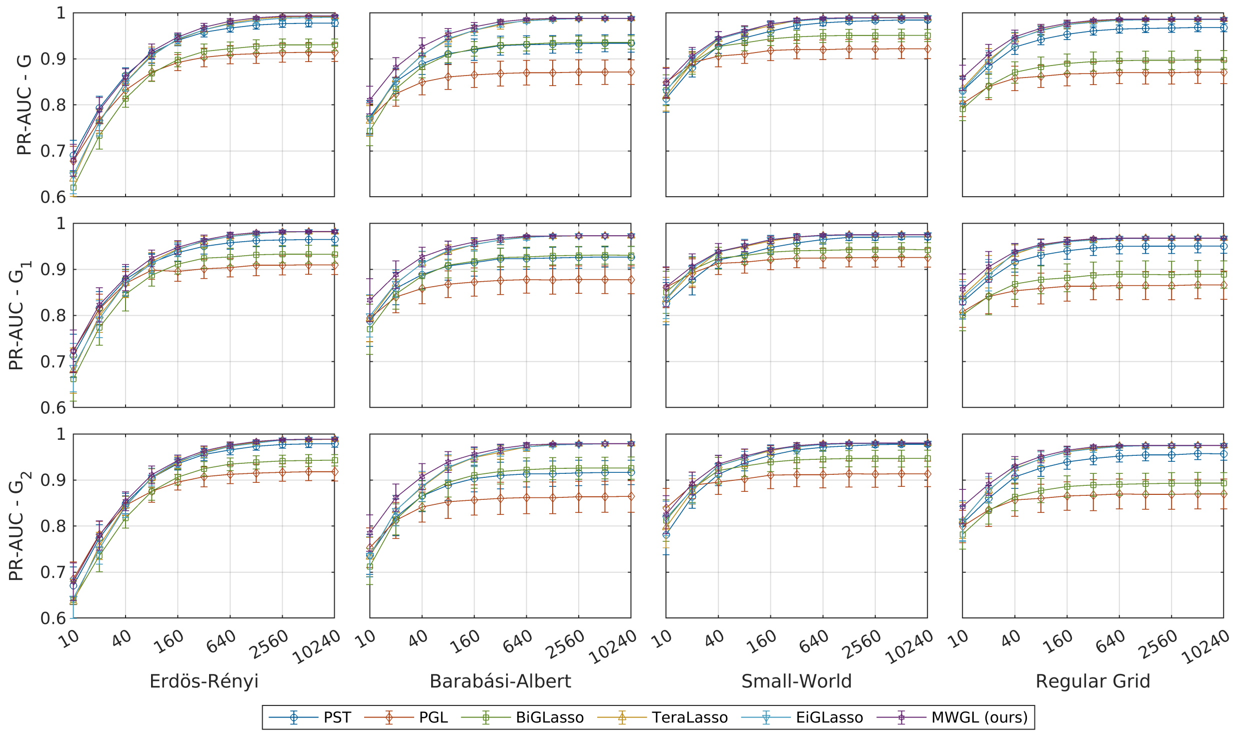


Figure 5: Comparison of different methods on synthetic data in various scenarios. Each sub-figure shows the PR-AUC of edge estimation as n increases.

B.2 Synthetic Experiments with Varying Factor Size

We now evaluate MWGL on synthetic data with fixed p but varying p_1 and p_2. Our main goal is to verify the convergence rate in 4 as a function of $\min(p_1, p_2)$, but we also compare MWGL with PGL and TeraLasso. We fix the size of product graphs to be $p = 256$ and set p_1 to be 4, 8, or 16, and p_2 to be 64, 32, and 16, respectively. We use the same graph models stated in Sec. 5.1 to generate factor graphs. For regular grids, we always set the width to 2 and the height correspondingly. We generate $n = 80$ graph signals and average the results across 50 realizations. Fig. 6 shows that MWGL again outperforms selected baselines and matches the theoretical results.

B.3 Graph Learning Comparison on Molene

We now compare our MWGL to PGL and TeraLasso, two methods that come close to MWGL on the synthetic data, on the Molene dataset across ranging regularization parameters. Fig. 7 shows the weighted adjacency matrices (negative off-diagonal precision matrices for TeraLasso) of the station graphs learned by these methods. First, notice that TeraLasso learns few negative conditional dependencies among weather stations. Indeed it is reasonable that the temperature of different locations does not depend negatively, which indicates that the attractive Laplacian constraints are suitable structural priors for the problem. Also, notice that only MWGL learns connected graphs with varying regularization, and neither PGL nor TeraLasso learns connected graphs when sparsity increases.

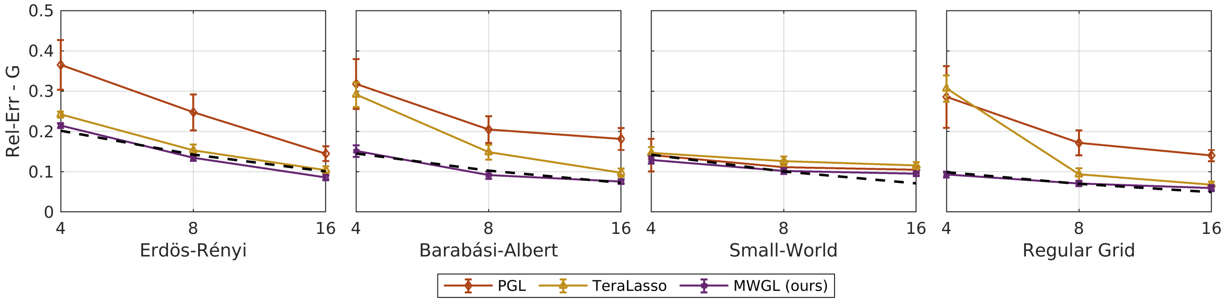


Figure 6: Comparison of different methods on synthetic data in various scenarios. Each sub-figure shows the trend of Rel-Err of the product or factor Laplacian matrices as $\min(p_1, p_2))$ increases. Black dash lines fit the theory in (14) to our results.

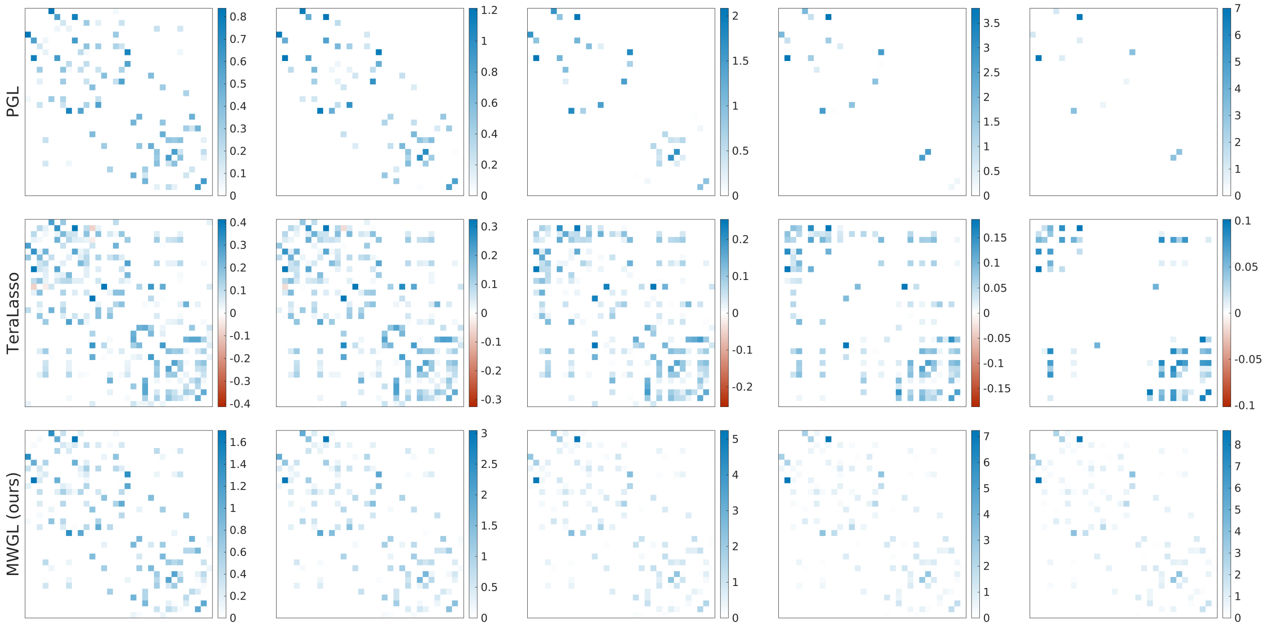


Figure 7: Comparing the learned station graph of PGL, TeraLasso, and MWGL (ours) on the Molene dataset with varying regularization. Laplacians are ordered with increasing sparsity from left to right.

References

Barik, S., Kalita, D., Pati, S., and Sahoo, G. (2018). Spectra of graphs resulting from various graph operations and products: a survey. *Special Matrices*, 6(1):323–342.

Einizade, A. and Sardouie, S. H. (2023). Learning product graphs from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*.

Fang, Y., Loparo, K. A., and Feng, X. (1994). Inequalities for the trace of matrix product. *IEEE Transactions on Automatic Control*, 39(12):2489–2490.

Greenewald, K., Zhou, S., and Hero III, A. (2019). Tensor graphical lasso (teralasso). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(5):901–931.

Hanson, D. L. and Wright, F. T. (1971). A bound on tail probabilities for quadratic forms in independent random variables. *The Annals of Mathematical Statistics*, 42(3):1079–1083.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation.

Rudelson, M. and Vershynin, R. (2013). Hanson-wright inequality and sub-gaussian concentration.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC press.

Segarra, S., Marques, A. G., Mateos, G., and Ribeiro, A. (2017). Network topology inference from spectral templates. *IEEE Trans. Signal Inf. Process*, 3(3):467–483.

Ying, J., de Miranda Cardoso, J. V., and Palomar, D. (2020). Nonconvex sparse graph learning under laplacian constrained graphical model. *Advances in Neural Information Processing Systems*, 33:7101–7113.

Ying, J., de Miranda Cardoso, J. V., and Palomar, D. (2021). Minimax estimation of laplacian constrained precision matrices. In *International Conference on Artificial Intelligence and Statistics*, pages 3736–3744. PMLR.