Learning Kronecker-Structured Graphs from Smooth Signals

Changhao Shi
University of California San Diego
cshi@ucsd.edu

Gal Mishne
University of California San Diego
gmishne@ucsd.edu

Abstract

Graph learning, or network inference, is a prominent problem in graph signal processing (GSP). GSP generalizes the Fourier transform to non-Euclidean domains, and graph learning is pivotal to applying GSP when these domains are unknown. With the recent prevalence of multi-way data, there has been growing interest in product graphs that naturally factorize dependencies across different ways. However, the types of graph products that can be learned are still limited for modeling diverse dependency structures. In this paper, we study the problem of learning a Kronecker-structured product graph from smooth signals. Unlike the more commonly used Cartesian product, the Kronecker product models dependencies in a more intricate, non-separable way, but posits harder constraints on the graph learning problem. To tackle this non-convex problem, we propose an alternating scheme to optimize each factor graph in turn and provide theoretical guarantees for its asymptotic convergence. The proposed algorithm is also modified to learn factor graphs of the strong product. We conduct experiments on synthetic and real-world graphs and demonstrate our approach's efficacy and superior performance compared to existing methods.

1 Introduction

GSP is a fast-growing field that extends classical signal processing (SP) to non-Euclidean domains [1, 2]. For a complex system, GSP studies the matrix representation of its graph abstraction. The spectral decomposition of these graph representations carries important geometric information, from which the Graph Fourier Transform (GFT) is established to analyze and process data on the graph.

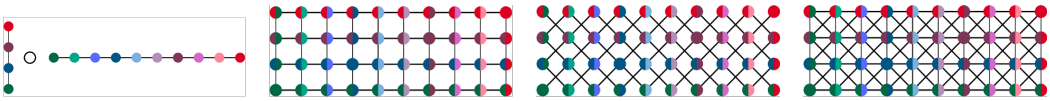
GSP finds its applications in plenty of fields [3–6], but a prominent problem even before applying GSP is that the graph abstraction of the studied system is frequently unobserved. Although constructing ad hoc graphs for specific applications may not be difficult, GSP resorts to a more principled method for learning these graphs from the nodal observations (signals). This data-driven methodology is called graph learning or network topology inference [7].

Graph learning imposes various prior assumptions on the observed data and solves for the graph that fits the best. Here, we focus on the smoothness assumption, which implies that the observed signals are smooth with respect to the graph of interest. The smoothness measurement is usually defined as a form of total variation of the graph signals and is related to the combinatorial graph Laplacian. One can then pose graph learning as an optimization problem regarding the Laplacian matrix.

With the prevalence of multi-way signals or tensors, there has been growing interest in extending GSP to these higher-order structures. The graph product prevails as a convenient tool since the factor graphs naturally capture the mode-wise dependencies of the data [8]. For example, the Cartesian graph product models a non-interactive, parallel composition of factor graphs and serves as the foundation of multi-way GSP [9]. However, other graph products are still under-explored in GSP.

The Kronecker graph product is a powerful model to simulate realistic graphs [10]. Fig. 1c shows an example of the Kronecker graph product. Unlike the Cartesian product, the Kronecker product wires factor graphs recursively to create a hierarchy with self-similarity. This model is shown to be useful for mimicking the network characteristics, such as degree distributions of real-world graphs.

, Learning Kronecker-Structured Graphs from Smooth Signals. *Proceedings of the Fourth Learning on Graphs Conference (LoG 2025)*, PMLR 338, Hybrid Event, December 10–12, 2025.



(a) Factor graphs. **(b)** Cartesian product. **(c)** Kronecker product. **(d)** Strong product

Figure 1: An example that compares the Cartesian, Kronecker, and strong graph products.

These beneficial properties make the Kronecker graph product a worthy candidate for modeling multi-dimensional structures in GSP [8]. Subsequently, how to learn rigorous Kronecker product graphs from the data emerges naturally as an interesting problem.

In this paper, we study the problem of learning the Kronecker product graph Laplacian from smooth multi-dimensional data. GSP has a probabilistic interpretation using the language of graphical models (GM), and graph learning from smooth signals boils down to the parameter estimation of the improper Gaussian Markov random field (IGMRF). We follow a similar route and formulate our problem as the penalized MLE of an IGMRF with Kronecker product constraints. As the problem is not jointly convex, we propose an algorithm that alternates between the optimization of each factor graph. We also provide theoretical results for the asymptotic convergence of the alternating algorithm, showing an improved convergence rate compared to when the product structure is not accounted for. Given that the strong graph product also bears a similar Kronecker product form, we also propose a variant of our algorithm to learn strong product graphs from smooth signals. We conduct experiments on synthetic and real-world graphs and demonstrate our approach's efficacy and superior performance compared to existing methods. The connections and differences between our method and related GSP and GM methods will also be discussed. To summarize our contributions:

- We are the first to consider the penalized MLE of Kronecker product graph Laplacian learning, and gain theoretical results on its asymptotic consistency, to the best of our knowledge.
- We propose a new algorithm to solve the penalized MLE and a variant of it to solve strong product graph Laplacian learning.
- We demonstrate that our approach outperforms existing GSP and GM methods on synthetic and real-world datasets.

Notations: 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 the Kronecker sum of two matrices, respectively. $\times$, $\square$, and $\boxtimes$ are used to denote the Kronecker product, the Cartesian product, and the strong product of two graphs, respectively. With abuse of notation, $\times$ also denotes the Cartesian product of two sets. For a node pair (v, u), $\sim$ denotes an edge connects them, and $\not\sim$ denotes non-connection. For matrix norms, $\|\cdot\|_F$ denotes the Frobenius norm, $\|\cdot\|_2$ the operator norm, and $\|\cdot\|_{1,\text{off}}$ the sum of the absolute values of all off-diagonal elements. For random variables, $\|\cdot\|_{\psi_2}$ denotes the sub-Gaussian norm. $(\,\cdot\,)_+$ denotes the non-negative projection. $[\,\cdot\,]_{I,J}$ denotes the sub-matrix of a $n \times m$ matrix at a subset of indices (I, J), where $I \subseteq \{1, 2, \ldots, n\}$ and $J \subseteq \{1, 2, \ldots, m\}$.

2 Background

2.1 Graph Representations

Consider an undirected, connected graph G with $|\mathcal{V}| = p$ vertices and $|\mathcal{E}|$ edges. A graph representation of G is a matrix that fully determines the topology of G. One of the most common graph representations is the weighted symmetric adjacency matrix $\mathbf{W} \in \mathbb{S}^p$. Each entry of the weight matrix $[\mathbf{W}]_{ij} = [\mathbf{W}]_{ji} \geq 0$ encodes the weight of 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. $\mathbf{W}_{ii} = 0$. Another important graph representation is the combinatorial graph Laplacian matrix $\mathbf{L}$. The Laplacian of the graph G is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ denotes the diagonal degree matrix where $[\mathbf{D}]_{ii} = \sum_j [\mathbf{W}]_{ij}$. The Laplacian matrix is

positive semi-definite by definition, i.e. $\mathbf{L} \in \mathbb{S}_+^p$, with the number of zero eigenvalues equal to the number of connected components in the graph. The Laplacian matrix plays a vital role in spectral graph theory, graph machine learning, and many other scientific fields [11].

Let $\mathbf{w} \in \mathbb{R}^{p(p-1)/2}$ denote the vectorization of the graph weights, where $[\mathbf{w}]_{i-j+\frac{1}{2}(j-1)(2p-j)} = [\mathbf{W}]_{ij}, \forall 1 \leq j \leq i \leq p$. By definition, $\mathbf{w}$ is also a graph representation. We then define the linear maps from this non-negative weight vector to its corresponding weighted adjacency matrix and combinatorial graph Laplacian, following [12]. These linear maps pave the way for our derivation since we will use different graph representations throughout the paper.

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

$$[\mathcal{A}\mathbf{w}]_{ij} = \begin{cases} -[\mathbf{w}]_{i-j+\frac{1}{2}(j-1)(2p-j)} & i > j, \\ [\mathcal{A}\mathbf{w}]_{ji} & i < j, \\ 0 & i = j. \end{cases}$$

Definition 2.2. 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

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

It is obvious that $\mathbf{W} = \mathcal{A}\mathbf{w}$. One can verify that $\mathcal{L}\mathbf{w}$ is a combinatorial graph Laplacian with weights $\mathbf{w}$. We then define their adjoint operators.

Definition 2.3. Define $\mathcal{A}^* : \mathbb{R}^{p\times p} \to \mathbb{R}^{p(p-1)/2}, \mathbf{Q} \mapsto \mathcal{A}^*\mathbf{Q}$ as the following

$$[\mathcal{A}^*\mathbf{Q}]_l = \frac{1}{2}([\mathbf{Q}]_{ij} + [\mathbf{Q}]_{ji}),\ l = i - j + \frac{1}{2}(j-1)(2p-j),\ i > j.$$

Definition 2.4. 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

$$[\mathcal{L}^*\mathbf{Q}]_l = [\mathbf{Q}]_{ii} - [\mathbf{Q}]_{ij} - [\mathbf{Q}]_{ji} + [\mathbf{Q}]_{jj},\ l = i - j + \frac{1}{2}(j-1)(2p-j),\ i > j.$$

2.2 Smoothness Prior in GSP and GM

We consider a vector-valued function $\boldsymbol{f} : \mathcal{V} \to \mathbb{R}^p$, which assigns a scalar value to each vertex of the graph. The combinatorial graph Laplacian induces a quadratic $\boldsymbol{f}^T\mathbf{L}\boldsymbol{f}$, also known as the Dirichlet energy. The Laplacian quadratic term measures the smoothness (variation) of $\boldsymbol{f}$ with respect to G, as $\boldsymbol{f}^T\mathbf{L}\boldsymbol{f} = \sum_{ij} [\mathbf{W}]_{ij}([\boldsymbol{f}]_i - [\boldsymbol{f}]_j)^2$ can be shown. Given n graph signals $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$, the inner product $\langle \mathbf{L}, \mathbf{S}\rangle = \mathrm{Tr}\,(\mathbf{LS})$ of the Laplacian and the sample covariance matrix (SCM) $\mathbf{S} = \frac{1}{n}\sum_{k=1}^n \mathbf{f}_k\mathbf{f}_k^T$ measures the overall smoothness of these signals with respect to the graph. GSP tackles graph learning by solving the penalized objective function

$$\min_{\mathbf{L}\in\Omega_\mathbf{L}} \left\{\langle \mathbf{L}, \mathbf{S}\rangle + \alpha h(\mathbf{L})\right\}, \tag{1}$$

where $\Omega_\mathbf{L}$ is the space of all combinatorial graph Laplacians:

$$\Omega_\mathbf{L} := \left\{\mathbf{L} \in \mathbf{S}_+^p \mid \mathbf{L}\mathbf{1} = \mathbf{0}, [\mathbf{L}]_{ij} = [\mathbf{L}]_{ji} \leq 0, \forall i \neq j\right\},$$

where $h(\mathbf{L})$ is a penalty term, and $\alpha > 0$ is a trade-off parameter. Minimizing $\langle \mathbf{L}, \mathbf{S}\rangle$ ensures signals $\{\mathbf{f}_k\}$ to vary smoothly on the inferred graph $\mathbf{L}$. The penalty term $h(\mathbf{L})$ prevents trivial solutions such as $\mathbf{L} = \mathbf{O}$, and it often encourages other structural properties such as sparsity.

GM approaches the graph learning problem from a different path. Consider an IGMRF $\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^\dagger)$, the penalized MLE is reminiscent of the graphical lasso

$$\min_{\mathbf{L}\in\Omega_\mathbf{L}} \left\{\langle \mathbf{L}, \mathbf{S}\rangle - \log\det{}^\dagger(\mathbf{L}) + \alpha\|\mathbf{L}\|_{1,\mathrm{off}}\right\}. \tag{2}$$

The additional ℓ_1 penalization promotes sparsity, and $\alpha > 0$ controls its strength. The Laplacian constraint is what makes (2) substantially different from covariance selection, since the solution spaces of these two problems are disjoint. Laplacian matrices in $\Omega_\mathbf{L}$ are singular with constant 0-eigenvectors. They are also attractive, only allowing positive conditional dependencies. Covariance selection solves for ordinary precision matrices which are non-singular and have both positive and negative dependencies. Also notice that one obtains (1) by substituting the penalization in (2) with $-\log\det{}^\dagger(\mathbf{L}) + \alpha\|\mathbf{L}\|_{1,\mathrm{off}}$. This demonstrates the connection between the GSP and GM formulations.

3 Kronecker Structured Graph Learning

3.1 Product Graph Learning

Consider two factor 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 $|\mathcal{V}_2| = p_2$. A graph product takes G_1 and G_2 and produces a larger graph G of $|\mathcal{V}| = |\mathcal{V}_1 \times \mathcal{V}_2| = p_1 p_2$ vertices. Two vertices (v_1, v_2) and (u_1, u_2) in the product graph G are connected iff some product-specific conditions are satisfied. An example is the Cartesian product $G = G_1 \square G_2$, where $(v_1, v_2) \sim (u_1, u_2)$ 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 $\mathbf{W}_1 \oplus \mathbf{W}_2$.

Although the Cartesian graph product is widely used for modeling non-interactive dependency structures, many applications desire more intricate, interactive structures [8]. As an example, in a time-vertex structure in which nodes 'interact' across time, such as the ones in neuroscience, communication, and traffic flows, there exist clear dependencies between neighboring nodes at adjacent time points. In this setting, the Cartesian product structure is over-simplified and incapable of modeling such dependencies, so we turn to other graph products.

We focus on the Kronecker and strong products, which are two other common options for modeling product graphs. The Kronecker product of G_1 and G_2 is denoted as $G = G_1 \times G_2$, where $(v_1, v_2) \sim (u_1, u_2)$ iff $v_1 \sim v_2 \wedge u_1 \sim u_2$. The weighted adjacency matrix of G is the Kronecker product of the factor weights $\mathbf{W}_1 \otimes \mathbf{W}_2$. Another graph product that produces even denser connectivity is the strong product. The strong product $G = G_1 \boxtimes G_2$ is defined as the union of the Kronecker and the Cartesian products. Thus the weighted adjacency matrix of the strong product graph is $\mathbf{W}_1 \otimes \mathbf{W}_2 + \mathbf{W}_1 \oplus \mathbf{W}_2$. Fig. 1 illustrates the aforementioned three graph products.

We now formulate the product graph learning problem using the Kronecker graph product. This formulation also enlightens strong product graph learning, as discussed later. 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 Kronecker product G from these nodal observations on G. Note that our argument can be generalized to more factors naturally, though not presented here.

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 $G = G_1 \times G_2$ we have $\mathbf{A} = \mathbf{A}_1 \otimes \mathbf{A}_2$, we derive the non-penalized product graph learning objective

$$\min_{\mathbf{w}_1, \mathbf{w}_2 \in \Omega_{\mathbf{w}}} \Big\{ \langle \mathbf{L}, \mathbf{S} \rangle - \log\det{}^{\dagger}(\mathbf{L}) \Big\}, \text{ s.t. } \mathbf{L} = \mathbf{D}_1 \otimes \mathbf{D}_2 - \mathbf{W}_1 \otimes \mathbf{W}_2 \tag{3}$$

It is worth emphasizing that $\mathbf{L} \neq \mathbf{L}_1 \otimes \mathbf{L}_2$ and thus $\log\det{}^{\dagger}(\mathbf{L}) \neq \log\det{}^{\dagger}(\mathbf{L}_1) + \log\det{}^{\dagger}(\mathbf{L}_2)$. This differentiates our problem from the MLE of matrix normal distributions [13]. Interestingly, except for the GSP merits, the Laplacian constraints also endow total positivity [14], a GM property that is not compatible with other Kronecker structured graphical models.

3.2 Kronecker Product Graphs

We propose the **K**ronecker **S**tructured **G**raph (Laplacian) **L**earning (**KSGL**) algorithm for solving (3). We formulate the penalized MLE (3) as

$$\min_{\mathbf{w}_1, \mathbf{w}_2 \geq \mathbf{0}} \Big\{ \langle \mathcal{A}\mathbf{w}_1 \otimes \mathcal{A}\mathbf{w}_2, \mathbf{K} \rangle + \alpha_1 \mathbf{w}_1^T \mathbf{1} + \alpha_2 \mathbf{w}_2^T \mathbf{1} - \log\det{}^{\dagger}(\mathcal{L}\mathcal{A}^*(\mathcal{A}\mathbf{w}_1 \otimes \mathcal{A}\mathbf{w}_2)) \Big\}, \tag{4}$$

with ℓ_1 penalization. Here $\mathbf{K} = \mathcal{A}\mathcal{L}^*\mathbf{S}$ denotes the pairwise square Euclidean distances of the signals. The absolute sign of the ℓ_1 norm of the sparsity penalty is redundant due to the non-negative constraints. KSGL then operates in an alternating scheme to solve $\mathbf{w}_1$ and $\mathbf{w}_2$. The algorithm starts with initialization $\mathbf{w}_1 = \frac{1}{p_1}\mathbf{1}_{\frac{p_1(p_1-1)}{2}}$ and $\mathbf{w}_2 = \frac{1}{p_2}\mathbf{1}_{\frac{p_2(p_2-1)}{2}}$. It then uses projected gradient descent to solve for one variable while keeping the other fixed until the stopping criteria are met. The update of $\mathbf{w}_1$ and $\mathbf{w}_2$ is given by

$$\begin{aligned} [\mathbf{w}_1^{(t+1)}]_{m_1} = \Big(& [\mathbf{w}_1^{(t)}]_{m_1} - \eta(\langle \mathcal{A}\mathbf{w}_2^{(t)}, [\mathbf{K}]_{I_1, J_1} \rangle - \\ & \langle \mathcal{A}\mathbf{w}_2^{(t)}, [\mathcal{A}\mathcal{L}^*(\mathbf{L}^{(t)} + \mathbf{J})^{-1}]_{I_1, J_1} \rangle + \alpha_1) \Big)_+, \end{aligned} \tag{5}$$

Algorithm 1 KSGL

Input: graph signals $\{\mathbf{X}_k\}$, parameters α, η
Output: factor graph weights $\mathbf{w}_1, \mathbf{w}_2$
Compute $\mathbf{S}$ and $\mathbf{K}$. Initialize $\mathbf{w}_1$ and $\mathbf{w}_2$.
repeat
  repeat
    Update $\mathbf{w}_1$ as in (5) for the Kronecker product or (8) for the strong product
  until convergence.
  repeat
    Update $\mathbf{w}_2$ as in (6) for the Kronecker product or (9) for the strong product
  until $\mathbf{w}_2$ convergence.
until $\mathbf{w}_1$ and $\mathbf{w}_2$ converge or maximum iterations.

$$[\mathbf{w}_2^{(t+1)}]_{m_2} = \Big([\mathbf{w}_2^{(t)}]_{m_2} - \eta(\langle \mathcal{A}\mathbf{w}_1^{(t+1)}, [\mathbf{K}]_{I_2,J_2}\rangle - \langle \mathcal{A}\mathbf{w}_1^{(t+1)}, [\mathcal{A}\mathcal{L}^*(\mathbf{L}^{(t')} + \mathbf{J})^{-1}]_{I_2,J_2}\rangle + \alpha_2)\Big)_+. \tag{6}$$

Here $m_1 = i - j + \frac{1}{2}(j-1)(2p_1 - j)$ and $m_2 = i - j + \frac{1}{2}(j-1)(2p_2 - j)$. The subsets $I_1 = \{(i-1)p_2+1, (i-1)p_2+2, \ldots, ip_2\}$ and $J_1 = \{(j-1)p_2+1, (j-1)p_2+2, \ldots, jp_2\}$ specify node pairs associated with $[\mathbf{w}_1]_{m_1}$, and similarly for the subsets $I_2 = \{i, p_2+i, \ldots, (p_1-1)p_2+i\}$ and $J_2 = \{j, p_2+j, \ldots, (p_1-1)p_2+j\}$. Alg. 1 summarizes the algorithm.

3.3 Strong Product Graphs

An alternative graph product with broad applications is the strong graph product. We now demonstrate how the strong product relates to the Kronecker product and how we can easily modify KSGL to learn strong product graphs. For factor graphs G_1 and G_2, consider adding self-loops to them and then taking the Kronecker product. The new product graph is also self-looped and its weighted adjacency matrix is $(\mathbf{W}_1 + \mathbf{I}_{p_1}) \otimes (\mathbf{W}_2 + \mathbf{I}_{p_2}) = \mathbf{W}_1 \otimes \mathbf{W}_2 + \mathbf{W}_1 \otimes \mathbf{I}_{p_2} + \mathbf{I}_{p_1} \otimes \mathbf{W}_2 + \mathbf{I}_p = (\mathbf{W}_1 \otimes \mathbf{W}_2 + \mathbf{W}_1 \oplus \mathbf{W}_2) + \mathbf{I}_p$. Removing the self-loops, we obtain exactly the strong product of G_1 and G_2. This relation helps us formulate the penalized MLE for learning strong product graphs based on (4)

$$\min_{\mathbf{w}_1,\mathbf{w}_2 \geq \mathbf{0}} \Big\{ \langle (\mathcal{A}\mathbf{w}_1 + \mathbf{I}_{p_1}) \otimes (\mathcal{A}\mathbf{w}_2 + \mathbf{I}_{p_2}), \mathbf{K}\rangle - \log\det{}^{\dagger}(\mathcal{L}\mathcal{A}^*((\mathcal{A}\mathbf{w}_1 + \mathbf{I}_{p_1}) \otimes (\mathcal{A}\mathbf{w}_2 + \mathbf{I}_{p_2}))) + \alpha_1 \mathbf{w}_1^T \mathbf{1} + \alpha_2 \mathbf{w}_2^T \mathbf{1} \Big\}. \tag{7}$$

Here we plug in the self-looped strong product adjacency matrix since the pairwise distances are all 0 on the $\mathbf{K}$ diagonal and $\mathcal{A}^*$ is also agnostic to its input diagonal values. Similarly, we use projected gradient descent to solve for $\mathbf{w}_1$ or $\mathbf{w}_2$ and then alternate between these two steps. The update of $\mathbf{w}_1$ and $\mathbf{w}_2$ is

$$[\mathbf{w}_1^{(t+1)}]_{m_1} = \Big([\mathbf{w}_1^{(t)}]_{m_1} - \eta(\langle \mathcal{A}\mathbf{w}_2^{(t)} + \mathbf{I}_{p_2}, [\mathbf{K}]_{I_1,J_1}\rangle - \langle \mathcal{A}\mathbf{w}_2^{(t)} + \mathbf{I}_{p_2}, [\mathcal{A}\mathcal{L}^*(\mathbf{L}^{(t)} + \mathbf{J})^{-1}]_{I_1,J_1}\rangle + \alpha_1)\Big)_+, \tag{8}$$

$$[\mathbf{w}_2^{(t+1)}]_{m_2} = \Big([\mathbf{w}_2^{(t)}]_{m_2} - \eta(\langle \mathcal{A}\mathbf{w}_1^{(t+1)} + \mathbf{I}_{p_1}, [\mathbf{K}]_{I_2,J_2}\rangle - \langle \mathcal{A}\mathbf{w}_1^{(t+1)} + \mathbf{I}_{p_1}, [\mathcal{A}\mathcal{L}^*(\mathbf{L}^{(t')} + \mathbf{J})^{-1}]_{I_2,J_2}\rangle + \alpha_2)\Big)_+. \tag{9}$$

4 Theoretical Results

Here we establish the statistical consistency and convergence rates for the penalized Kronecker product graph Laplacian estimator as in (4). We first make assumptions regarding the true underlying graph we were to estimate:

(A1) Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be the support set of the true factor graphs. We assume the graphs are sparse and the cardinality of their supports is upper bounded by $|\mathcal{S}_1| \leq s_1 p_1$ and $|\mathcal{S}_2| \leq s_2 p_2$.

(A2) Let $(d_{1,\min}, d_{1,\max})$ and $(d_{2,\min}, d_{2,\max})$ be the minimum and maximum degrees of the true factor graphs. We assume these degrees are bounded away from 0 and ∞ by a constant $d > 1$, such that $\frac{1}{d} \leq d_{1,\min} \leq d_{1,\max} \leq d$ and $\frac{1}{d} \leq d_{2,\min} \leq d_{2,\max} \leq d$.

(A3) Let $\{0, \lambda_2, \ldots, \lambda_p\}$ be the eigenvalues of the true product graph Laplacian in a non-decreasing order. We assume these eigenvalues are bounded away from 0 and ∞ by a constant $z > 1$, such that $\frac{1}{z} \leq \lambda_2 < \lambda_p \leq z$.

These assumptions are common in high-dimensional statistics. They also imply that the product and factor graphs are connected graphs. With the above assumptions, our first theorem states that a solution to the MLE problem always exists. Proofs of all the theorems can be found in the supplement.

Theorem 4.1 (Existence of MLE)**.** *The penalized negative log-likelihood of Kronecker product graph Laplacian learning as in* (4) *is lower-bounded, and there exists at least one global minimizer as the solution of the penalized MLE.*

Our proof largely follows [15, 16], which shows that the objective function is lower-bounded and the minimizer is achievable. However, note that since the original problem is not jointly convex, the solution is not unique. In fact, a set of solution $(\mathbf{w}_1^*, \mathbf{w}_2^*)$ is not identifiable to the Kronecker graph product $\mathcal{A}\mathbf{w}_1 \otimes \mathcal{A}\mathbf{w}_2$ since $\forall a > 0, a\mathbf{w}_1^*, \frac{1}{a}\mathbf{w}_2^*$ is also a solution. Nevertheless, our Theorem 4.2 states that the alternating optimization enjoys a unique solution in each sub-problem.

Theorem 4.2 (Uniqueness of MLE)**.** *The objective function of the penalized MLE is bi-convex with respect to each factor graph, and a global minimizer for each sub-problem uniquely exists.*

The proof shows that when one of the factors is held fixed, optimizing the other factor becomes a convex problem. We then show that the Kronecker product graph Laplacian learned by KSGL is asymptotically consistent.

Corollary 4.3. *Suppose assumptions (A1)-(A3) hold for the true factor graphs. Then with sufficiently large n and proper penalty α_1 and α_2, the Frobenius errors of the minimizers $\widehat{\mathbf{L}}_1 = \mathcal{L}(\widehat{\mathbf{w}}_\mathbf{1})$ and $\widehat{\mathbf{L}}_2 = \mathcal{L}(\widehat{\mathbf{w}}_\mathbf{2})$ are bounded by*

$$\|\widehat{\mathbf{L}}_1 - \mathbf{L}_1^*\|_F \leq c_1 \sqrt{\frac{s_1 p_1 \log p}{n p_2}}, \qquad (10) \qquad\qquad \|\widehat{\mathbf{L}}_2 - \mathbf{L}_2^*\|_F \leq c_2 \sqrt{\frac{s_2 p_2 \log p}{n p_1}}. \qquad (11)$$

with high probability.

Corollary 4.3 proves that the solution of a sub-problem converges to the ground truth when the other factor is bounded. This helps use induction to prove the consistency of (4).

Theorem 4.4 (High-dimensional consistency)**.** *Suppose assumptions (A1)-(A3) hold for the true factor graphs. Then with sufficiently large n, proper penalty α_1 and α_2, and Corollary 4.3 the minimizer $\widehat{\mathbf{L}} = \mathcal{L}(\widehat{\mathbf{w}})$ of the penalized MLE as in* (4) *is asymptotically consistent to the true Laplacian $\mathbf{L}^*$, and the Frobenius error is bounded by*

$$\|\widehat{\mathbf{L}} - \mathbf{L}^*\|_F \leq c \sqrt{\frac{(p_1 + p_2) \log p}{n}}. \qquad (12)$$

with high probability.

Theorem 4.4 shows that the learned product graph Laplacian converges to the true Laplacian asymptotically under mild conditions. The final error depends on the norms of initialization. Compared with the IGMRF convergence rate from [15], KSGL converges faster by a factor of $\sqrt{\frac{p_1 p_2}{p_1 + p_2}}$ with similar probability. This shows how leveraging the product structure prior benefits graph learning. The improvement of the convergence rate is similar to the ones in [17].

5 Related Work

Smooth Graph Learning. Learning graph Laplacian matrices from smooth signals has been studied extensively in GSP [18–23]. These papers focus on the Laplacian quadratic terms, which correspond to the Dirichlet energy of the signals. Thanou et al. [24] model the smoothness differently, using the

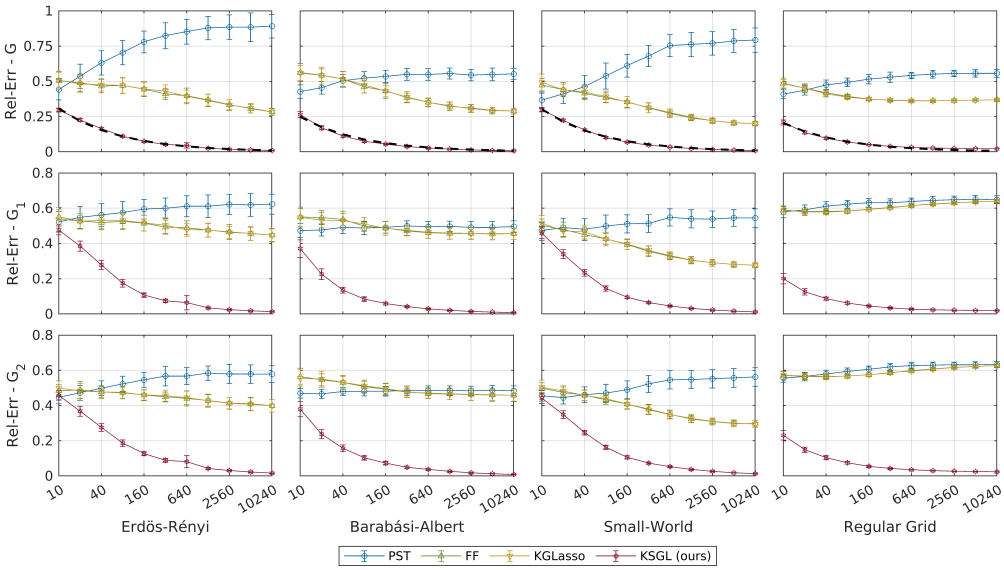


Figure 2: Comparison of different methods on various synthetic Kronecker product graphs and signals. Each sub-figure shows the trend of Rel-Err of the product (top row) or factor (middle and bottom rows) Laplacian matrices as n increases. Black dash lines fit the theory in (12) to KSGL.

heat diffusion process. Pasdeloup et al. [25] propose to learn the normalized graph Laplacian matrix instead of the combinatorial Laplacian. The weighted adjacency matrix has also been widely studied as a different graph representation [26, 27].

Product Graph Learning. Learning product graphs amounts to posing structural constraints on graph learning. Previous work mainly focuses on Cartesian product graphs [28–32]; few have studied other products such as the Kronecker [28, 31]. Lodhi and Bajwa [28] proposed to learn the factor graphs under the trace constraints; Einizade and Sardouie [31] posited that an accurate eigenbasis estimation of the factor graph shift operator (GSO) is known and solved for the eigenvalues. However, these methods either do not learn the combinatorial graph Laplacian, rely on different assumptions, or fall short of their theoretical properties.

Matrix Variate Distributions. It is well-known that the generative process of smooth graph signals can be modeled as a GMRF with a Laplacian precision matrix [33]. Therefore, methods for covariance selection [34–37] that aim to learn sparse precision matrices, such as the graphical lasso [38], often serve as additional baselines of graph learning methods. However, these methods do not learn a rigorous combinatorial graph Laplacian. The matrix variate normal distribution [39, 40] can be seen as a generalization of the GMRF to multi-way signals. The covariance matrices, and thus the precision matrices, are endowed with a Kronecker product structure, and the graphical lasso algorithm has been extended to learn these Kronecker graphical models [13, 17, 41–43]. Other matrix variate distributions replace the Kronecker product structure with the Kronecker sum [44–47], leading to Cartesian product graphs. While these graphical lasso methods endow various forms of the Kronecker product, none of these learn precision matrices are Laplacian, thus not appropriate for use in GSP.

6 Experiments

6.1 Synthetic Graphs

Since the ground truth graphs are often unavailable or not defined in real-world problems, we first evaluate our methods on synthetic signals where the underlying graph to be estimated is known. We follow [32] to generate factor graphs using the graph models below

(1) Erdős-Rényi model with probability $p = 0.3$;

(2) Barabási-Albert model with preferential attachment $m = 2$ and $m_0 = 2$ initial nodes;

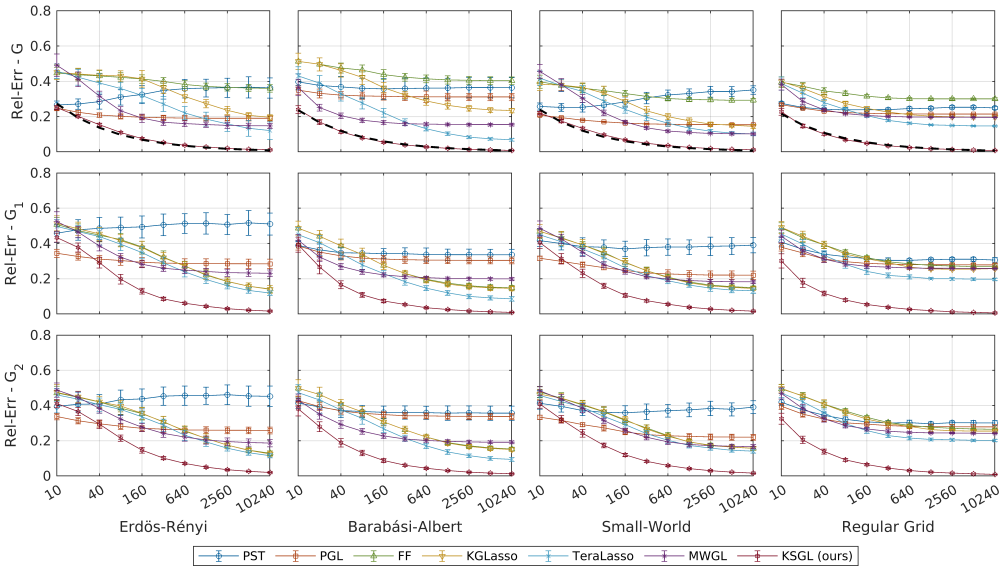


Figure 3: Comparison of different methods on various synthetic strong product graphs and signals. Each sub-figure shows the trend of Rel-Err of the product (top row) or factor (middle and bottom rows) Laplacian matrices as n increases. Black dash lines fit the theory in (12) to KSGL.

(3) Watts-Strogatz small-world model, where a chain graph of $d = 2$ is rewired with $p = 0.1$;

(4) and regular grids.

We set the number of nodes to $p_1 = 20$ and $p_2 = 25$ for each factor, and the dimensions of the regular grids are 4×5 and 5×5. To obtain weighted graphs, we randomly sample a weight from a uniform distribution $\mathcal{U}(0.1, 2)$ for each edge. We then generate the signals from the IGMRF process $\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^\dagger)$, where $\mathbf{L}$ is the Laplacian of the Kronecker product graph or the strong product graph. The goal of graph learning is to recover the underlying weighted graphs from the signals, where we vary the number of the signals $n = 10 \times 2^r, r \in \{0, 1, \ldots, 10\}$.

We create 50 realizations for each graph and dataset size and report the mean and standard deviation of the selected metrics: the relative error (Rel-Err) of the Laplacian and the area under the precision-recall curve (PR-AUC) of edge prediction. The former Rel-Err computes the relative Frobenius error of the learned factor and product graph Laplacian matrices to their ground truth counterparts. To eliminate the ambiguity of the learned factor graphs, we normalize the graph Laplacian matrices by their cardinality $\frac{p\mathbf{L}}{\mathrm{Tr}(\mathbf{L})}$ before computing the relative error. The latter PR-AUC considers the binary prediction of the ground truth edge patterns. We choose PR-AUC over ROC-AUC since the two classes are highly imbalanced (edge versus no edge).

We evaluate KSGL against three competing methods that model the Kronecker structures: the PST (Product Spectral Template) method [31], the FF (Flip-Flop) method [13], and the Kronecker Graphical Lasso (KGLasso) method [17]. PST is a GSP method that extracts the eigenvectors of factor GSOs from the signal covariance. FF solves the MLE of matrix normal distributions and KGLasso adds sparsity constraints to that, both of which fall into the GM category. For the strong product experiments, since the strong graph product is the union of the Kronecker and Cartesian graph product, we add Cartesian product graph learning methods for comparison: the PGL (Product Graph Learning) method [30], the TeraLasso (Tensor Graphical Lasso) method [45], and the MWGL (Multi-Way Graph Learning) method [16]. We follow the common grid-search procedure in each setting to select the best-performing hyper-parameters for each method.

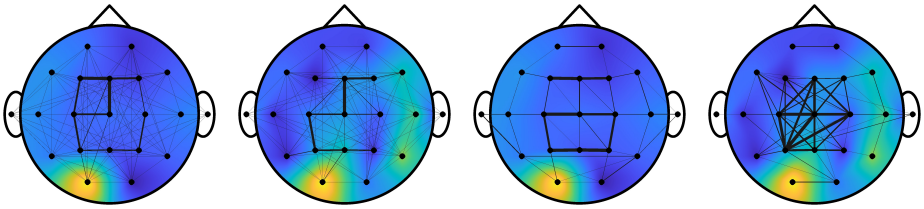
Fig. 2 shows the trend of Rel-Err as the number of signals increases in different settings. We provide the PR-AUC results in Appendix C. As we can see, KSGL outperforms the competing methods in every setting. The Rel-Err of KSGL converges to 0 as the number of signals increases and its trend validates our theoretical results (black dash lines, top row). PST does not perform well because the spectral templates cannot be estimated accurately but only roughly approximated even with a large

number of signals (see more details in Appendix B). FF and KGLasso also underperform KSGL because of the inherent model mismatch, which is that the Laplacian of a Kronecker product graph is not the Kronecker product of the factor Laplacians. Another reason is that the precision matrices learned by FF and KGLasso are not Laplacian. Their similar performance also shows that adding sparsity constraints to the wrong assumption does not benefit Kronecker product graph learning.

Fig. 3 shows the Rel-Err results of strong product graph learning. Again KSGL behaves advantageously in almost every setting. The only exception is that PGL performs better in low data regimes on Erdős-Rényi and Watts-Strogatz small-world graphs, but it fails to deliver as the number of signals increases due to the model mismatch. Among other competing methods, TeraLasso and MWGL perform well, but they still fall behind KSGL by a margin. Note that we do not include PGL, TeraLasso, and MWGL in the Kronecker product experiments since these Cartesian product methods are expected to fail in these settings. We verify this using Erdős-Rényi graphs in Appendix C.

6.2 EEG Data

We now evaluate KSGL on real EEG recordings [48]. The EEG data are collected from epileptic patients using the 10-20 electrode system. The signals from 21 scalp electrodes are divided into 1-second segments, and we sub-sample the signals to get 50 samples per segment. Each segment is also annotated to indicate the occurrence and the types of seizures. This results in 21×50 multi-way signals of different categories from multiple patients. Our goal is to learn a graph of brain regions (electrodes) and a graph of time from these multi-way signals.



(a) MWGL - normal **(b)** MWGL - seizure **(c)** KSGL - normal **(d)** KSGL - seizure

Figure 4: The brain connectivity inferred by MWGL and KSGL. Nodes reflect the actual electrode positions in the 10-20 system. The background color shows the mean EEG activity of each status.

Because brain connectivity varies dramatically across individuals, we pick the EEG of a single patient to evaluate KSGL. This 7-year-old male patient had several complex partial seizures in the central cortical area (Cz, C3, C4), but also went through multiple seizures that are not visible on EEG. We apply KSGL to learn strong product graphs from his normal EEGs and deceptive epileptic EEGs then compare the results with those of MWGL. Fig. 4 shows the brain connectivity graphs learned by MWGL and KSGL. As we can see, although the seizures are not obvious from the mean signal amplitude as expected, KSGL learns different connectivity patterns for these 2 statuses. In particular, KSGL learns denser connectivity around the central cortical area, matching the known lesion. On the other hand, MWGL learns similar brain connectivity patterns from normal and epileptic EEGs. Also note that the brain connectivity graphs learned by KSGL are sparser than those by MWGL, suggesting that the strong product suits this dataset better than the Cartesian product. Additional results of this patient and other patients are shown in Appendix D.

7 Conclusions

In this paper, we focus on graph learning, a classical problem in GSP, and extend it to learning Kronecker product structures from multi-way signals. We propose new algorithms for learning Kronecker and strong product graphs from smooth graph signals and evaluate their performance on both synthetic and real-world datasets. Our experiments show that the proposed KSGL methods outperform competing GSP and GM methods. We also investigate the theoretical aspects of the Kronecker algorithm and show that the solution of the penalized MLE converges to the true graph Laplacian asymptotically. Our results also prove that KSGL converges faster than general graph learning where the product structures are ignored. In the future, we intend to complete the theory for the strong product and improve the scalability of KSGL.

References

[1] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 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, 2013. 1

[2] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106 (5):808–828, 2018. 1

[3] Stephen P Borgatti, Ajay Mehra, Daniel J Brass, and Giuseppe Labianca. Network analysis in the social sciences. *science*, 323(5916):892–895, 2009. 1

[4] Georgios A Pavlopoulos, Maria Secrier, Charalampos N Moschopoulos, Theodoros G Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G Bagos. Using graph theory to analyze biological networks. *BioData mining*, 4:1–27, 2011.

[5] Danielle S Bassett and Olaf Sporns. Network neuroscience. *Nature neuroscience*, 20(3): 353–364, 2017.

[6] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020. 1

[7] Gonzalo Mateos, Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Process. Mag.*, 36(3):16–43, 2019. 1

[8] Aliaksei Sandryhaila and Jose MF Moura. 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, 2014. 1, 2, 4

[9] Jay S Stanley, Eric C Chi, and Gal Mishne. Multiway graph signal processing on tensors: Integrative analysis of irregular geometries. *IEEE signal processing magazine*, 37(6):160–173, 2020. 1

[10] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: an approach to modeling networks. *Journal of Machine Learning Research*, 11(2), 2010. 1

[11] Russell Merris. Laplacian matrices of graphs: a survey. *Linear algebra and its applications*, 197:143–176, 1994. 3

[12] Sandeep Kumar, Jiaxi Ying, José Vinícius de Miranda Cardoso, and Daniel P Palomar. A unified framework for structured graph learning via spectral constraints. *JMLR*, 21(22):1–60, 2020. 3

[13] Pierre Dutilleul. The mle algorithm for the matrix normal distribution. *Journal of statistical computation and simulation*, 64(2):105–123, 1999. 4, 7, 8

[14] Steffen Lauritzen, Caroline Uhler, and Piotr Zwiernik. Maximum likelihood estimation in gaussian models under total positivity. 2019. 4

[15] Jiaxi Ying, José Vinícius de Miranda Cardoso, and Daniel Palomar. Minimax estimation of laplacian constrained precision matrices. In *International Conference on Artificial Intelligence and Statistics*, pages 3736–3744. PMLR, 2021. 6

[16] Changhao Shi and Gal Mishne. Graph laplacian learning with exponential family noise. *arXiv preprint arXiv:2306.08201*, 2023. 6, 8

[17] Theodoros Tsiligkaridis, Alfred O Hero III, and Shuheng Zhou. On convergence of kronecker graphical lasso algorithms. *IEEE transactions on signal processing*, 61(7):1743–1755, 2013. 6, 7, 8

[18] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. Learning Laplacian matrix in smooth graph signal representations. *IEEE Trans. Signal Process.*, 64(23):6160–6173, 2016. 6

[19] Vassilis Kalofolias. How to learn a graph from smooth signals. In *AISTATS*, pages 920–929. PMLR, 2016.

[20] Sundeep Prabhakar Chepuri, Sijia Liu, Geert Leus, and Alfred O Hero. Learning sparse graphs under smoothness prior. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6508–6512. IEEE, 2017.

[21] Hilmi E Egilmez, Eduardo Pavez, and Antonio Ortega. Graph learning from data under Laplacian and structural constraints. *IEEE J. Sel. Topics Signal Process.*, 11(6):825–841, 2017.

[22] Licheng Zhao, Yiwei Wang, Sandeep Kumar, and Daniel P Palomar. Optimization algorithms for graph laplacian estimation via admm and mm. *IEEE Transactions on Signal Processing*, 67 (16):4231–4244, 2019.

[23] Andrei Buciulea, Samuel Rey, and Antonio G Marques. Learning graphs from smooth and graph-stationary signals with hidden variables. *IEEE Transactions on Signal and Information Processing over Networks*, 8:273–287, 2022. 6

[24] Dorina Thanou, Xiaowen Dong, Daniel Kressner, and Pascal Frossard. Learning heat diffusion graphs. *IEEE Trans. Signal Inf. Process*, 3(3):484–499, 2017. 6

[25] Bastien Pasdeloup, Vincent Gripon, Grégoire Mercier, Dominique Pastor, and Michael G Rabbat. Characterization and inference of graph diffusion processes from observations of stationary signals. *IEEE Trans. Signal Inf. Process*, 4(3):481–496, 2017. 7

[26] Santiago Segarra, Antonio G Marques, Gonzalo Mateos, and Alejandro Ribeiro. Network topology inference from spectral templates. *IEEE Trans. Signal Inf. Process*, 3(3):467–483, 2017. 7

[27] Rasoul Shafipour, Santiago Segarra, Antonio G Marques, and Gonzalo Mateos. Identifying the topology of undirected networks from diffused non-stationary graph signals. *IEEE Open Journal of Signal Processing*, 2:171–189, 2021. 7

[28] Muhammad Asad Lodhi and Waheed U Bajwa. Learning product graphs underlying smooth graph signals. *arXiv preprint arXiv:2002.11277*, 2020. 7

[29] Sai Kiran Kadambari and Sundeep Prabhakar Chepuri. Learning product graphs from multidomain signals. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5665–5669, 2020. doi: 10.1109/ICASSP40776.2020.9054679.

[30] Sai Kiran Kadambari and Sundeep Prabhakar Chepuri. Product graph learning from multidomain data with sparsity and rank constraints. *IEEE Transactions on Signal Processing*, 69: 5665–5680, 2021. 8, 21

[31] Aref Einizade and Sepideh Hajipour Sardouie. Learning product graphs from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*, 2023. 7, 8, 21

[32] Changhao Shi and Gal Mishne. Learning cartesian product graphs with laplacian constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 2521–2529. PMLR, 2024. 7, 21

[33] Cha Zhang, Dinei Florêncio, and Philip A Chou. Graph signal processing-a probabilistic framework. *Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2015-31*, 2015. 7

[34] Arthur P Dempster. Covariance selection. *Biometrics*, pages 157–175, 1972. 7

[35] Onureena Banerjee, Laurent El Ghaoui, Alexandre d'Aspremont, and Georges Natsoulis. Convex optimization techniques for fitting sparse gaussian graphical models. In *Proceedings of the 23rd international conference on Machine learning*, pages 89–96, 2006.

[36] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[37] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008. 7

[38] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008. 7

[39] A Philip Dawid. Some matrix-variate distribution theory: notational considerations and a bayesian application. *Biometrika*, 68(1):265–274, 1981. 7

[40] Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*, volume 104. CRC Press, 1999. 7

[41] Karl Werner, Magnus Jansson, and Petre Stoica. On estimation of covariance matrices with kronecker product structure. *IEEE Transactions on Signal Processing*, 56(2):478–491, 2008. 7

[42] Yi Zhang and Jeff Schneider. Learning multiple tasks with a sparse matrix-normal penalty. *Advances in neural information processing systems*, 23, 2010.

[43] Chenlei Leng and Cheng Yong Tang. Sparse matrix graphical models. *Journal of the American Statistical Association*, 107(499):1187–1200, 2012. 7

[44] Alfredo Kalaitzis, John Lafferty, Neil D Lawrence, and Shuheng Zhou. The bigraphical lasso. In *International Conference on Machine Learning*, pages 1229–1237. PMLR, 2013. 7

[45] Kristjan Greenewald, Shuheng Zhou, and Alfred Hero III. Tensor graphical lasso (teralasso). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(5):901–931, 2019. 8, 21

[46] Yu Wang, Byoungwook Jang, and Alfred Hero. The sylvester graphical lasso (syglasso). In *International Conference on Artificial Intelligence and Statistics*, pages 1943–1953. PMLR, 2020.

[47] Jun Ho Yoon and Seyoung Kim. Eiglasso for scalable sparse kronecker-sum inverse covariance estimation. *The Journal of Machine Learning Research*, 23(1):4733–4771, 2022. 7

[48] Wassim Nasreddine. Epileptic eeg dataset. https://data.mendeley.com/datasets/5pc2j46cbc, 2021. 9

[49] Jiaxi Ying, José Vinícius de Miranda Cardoso, and Daniel Palomar. Nonconvex sparse graph learning under laplacian constrained graphical model. *Advances in Neural Information Processing Systems*, 33:7101–7113, 2020. 15

[50] David Lee Hanson and Farroll Tim Wright. A bound on tail probabilities for quadratic forms in independent random variables. *The Annals of Mathematical Statistics*, 42(3):1079–1083, 1971. 16

[51] Mark Rudelson and Roman Vershynin. Hanson-wright inequality and sub-gaussian concentration. 2013. 16

[52] Hiroki Sayama. Estimation of laplacian spectra of direct and strong product graphs. *Discrete Applied Mathematics*, 205:160–170, 2016. 21

[53] Milan Bašić, Branko Arsić, and Zoran Obradović. Another estimation of laplacian spectrum of the kronecker product of graphs. *Information Sciences*, 609:605–625, 2022. 21

A Proof of Main Theorems

A.1 Proof of Theorem 4.1

Proof. Given $\mathcal{A}\mathbf{w} = \mathcal{A}\mathbf{w}_1 \otimes \mathcal{A}\mathbf{w}_2$, we now prove that the global minimizer of the penalized MLE

$$\min_{\mathbf{w}_1,\mathbf{w}_2 \geq \mathbf{0}} \left\{ \langle \mathcal{L}\mathbf{w}, \mathbf{S} \rangle - \log\det{}^{\dagger}(\mathcal{L}\mathbf{w}) + \alpha_1 \|\mathbf{w}_1\|_1 + \alpha_2 \|\mathbf{w}_2\|_1 \right\}, \tag{13}$$

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

$$\Omega_{\mathbf{w}_1,\mathbf{w}_2} := \{(\mathbf{w}_1, \mathbf{w}_2) | \mathbf{w}_1 \geq \mathbf{0}, \mathcal{L}\mathbf{w}_1 + \mathbf{J}_{p_1} \in \mathbb{S}^{p_2}_{++}, \mathbf{w}_2 \geq \mathbf{0}, \mathcal{L}\mathbf{w}_2 + \mathbf{J}_{p_2} \in \mathbb{S}^{p_2}_{++}\}, \tag{14}$$

where $\mathbf{J}_p = \frac{1}{p}\mathbf{1}_p\mathbf{1}_p^T$ and we have $\log\det{}^{\dagger}(\mathcal{L}\mathbf{w}) = \log\det(\mathcal{L}\mathbf{w} + \mathbf{J}_p)$. The conditions $\mathcal{L}\mathbf{w}_1 + \mathbf{J}_{p_1} \in \mathbb{S}^{p_1}_{++}$ and $\mathcal{L}\mathbf{w}_2 + \mathbf{J}_{p_2} \in \mathbb{S}^{p_2}_{++}$ constrain that G_1 and G_2 are connected. Let $\{0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_p\}$ be the eigenvalues of $\mathbf{L} = \mathcal{L}\mathbf{w}$. We first consider the original MLE and bound the negative log-likelihood $Q(\mathbf{w}_1, \mathbf{w}_2)$ when $\alpha_1 = \alpha_2 = 0$

$$\langle \mathcal{L}\mathbf{w}, \mathbf{S} \rangle - \log\det{}^{\dagger}(\mathcal{L}\mathbf{w}) \tag{15}$$

$$= \langle \mathcal{L}\mathbf{w}, \mathbf{S} \rangle - \log(\prod_{k=2}^{p} \lambda_k) \tag{16}$$

$$\geq \langle \mathcal{L}\mathbf{w}, \mathbf{S} \rangle - (p-1)\log(\sum_{k=1}^{p} \lambda_k) + (p-1)\log(p-1) \tag{17}$$

$$= \langle \mathcal{L}^*\mathbf{S}, \mathbf{w} \rangle - (p-1)\log(p\|\mathbf{w}\|_1) + (p-1)\log(\frac{p-1}{2}) \tag{18}$$

$$\geq \min(\mathcal{L}^*\tilde{\mathbf{S}})p\|\mathbf{w}\|_1 - (p-1)\log(p\|\mathbf{w}\|_1) + (p-1)\log(\frac{p-1}{2}), \tag{19}$$

where $[\tilde{\mathbf{S}}]_{i,j} = \frac{1}{p}[\mathbf{S}]_{i,j}$. (18) is attributed to the fact that the summation of eigenvalues equals the trace of the Laplacian. Define the function

$$q(t) = \min(\mathcal{L}^*\tilde{\mathbf{S}})t - (p-1)\log(t) + (p-1)\log(\frac{p-1}{2}). \tag{20}$$

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

$$Q(\mathbf{w}_1, \mathbf{w}_2) \geq q(p\|\mathbf{w}\|_1) \geq (p-1)(1 + \log(\frac{\min(\mathcal{L}^*\tilde{\mathbf{S}})}{2})). \tag{21}$$

We then notice that $q(t) \to \infty$ when $t \to \infty$. This is followed by $Q(\mathbf{w}_1, \mathbf{w}_2)$ being coercive.

Now consider the penalized MLE. When the penalization $\alpha_1 > 0$ and $\alpha_2 > 0$, the penalized MLE $Q(\mathbf{w}_1, \mathbf{w}_2)$ is still lower-bounded. Since

$$\mathbf{w}_1 \to \infty \mid \mathbf{w}_2 \to \infty \;\rightsquigarrow\; Q(\mathbf{w}_1, \mathbf{w}_2) \to \infty, \tag{22}$$

the penalized MLE $Q(\mathbf{w}_1, \mathbf{w}_2)$ is also coercive. Note that this only holds when we penalize the factor graphs. The penalized MLE wouldn't be coercive if the ℓ-1 penalization is on the product graph.

The above argument indicates that a global minimizer exists in $\mathrm{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2})$, and now we show that it exist in $\Omega_{\mathbf{w}_1,\mathbf{w}_2}$. Since the open boundaries $\mathrm{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2}) \setminus \Omega_{\mathbf{w}_1,\mathbf{w}_2}$ 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 $\mathrm{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2}) \setminus \Omega_{\mathbf{w}_1,\mathbf{w}_2}$ 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

$$\{(\mathbf{w}_1, \mathbf{w}_2) | \det(\mathcal{L}\mathbf{w}_1 + \mathbf{J}_{p_1}) = 0 \vee \det(\mathcal{L}\mathbf{w}_2 + \mathbf{J}_{p_2}) = 0\}. \tag{23}$$

Since for the Kronecker product, any factor graph being disconnected leads to the product graph being disconnected. Therefore,

$$(\mathbf{w}_1, \mathbf{w}_2) \in \mathrm{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2}) \setminus \Omega_{\mathbf{w}_1,\mathbf{w}_2} \;\rightsquigarrow\; \log\det(\mathcal{L}\mathbf{w} + \mathbf{J}_p) = -\infty \;\rightsquigarrow\; Q(\mathbf{w}_1, \mathbf{w}_2) \to \infty. \tag{24}$$

This shows that any global minimizer over $\mathrm{cl}(\Omega_{\mathbf{w}_1,\mathbf{w}_2})$ do not lie on those open boundaries, therefore (13) has at least a global minimizer in $\Omega_{\mathbf{w}_1,\mathbf{w}_2}$ so long as $\min(\mathcal{L}^*\tilde{\mathbf{S}}) > 0$, which almost surely holds with probability 1. □

A.2 Proof of Theorem 4.2

Proof. $Q(\mathbf{w}_1, \mathbf{w}_2)$ is not jointly convex on $\mathbf{w}_1$ and $\mathbf{w}_2$, but it is bi-convex with respect to each separate variable. This means the MLE objective is convex with respect to $\mathbf{w}_1$ when $\mathbf{w}_2$ is fixed, and also convex with respect to $\mathbf{w}_2$ when $\mathbf{w}_1$ is fixed. Define the feasible set of $\mathbf{w}_1$ and $\mathbf{w}_2$ as

$$\Omega_{\mathbf{w}_1} := \{\mathbf{w}_1 | \mathbf{w_1} > \mathbf{0}, \mathcal{L}\mathbf{w}_1 + \mathbf{J}_{p_1} \in \mathbb{S}^{p_1}_{++}\} \tag{25}$$

$$\Omega_{\mathbf{w}_2} := \{\mathbf{w}_2 | \mathbf{w_2} > \mathbf{0}, \mathcal{L}\mathbf{w}_2 + \mathbf{J}_{p_2} \in \mathbb{S}^{p_2}_{++}\}. \tag{26}$$

and we have $\Omega_{\mathbf{w}_1,\mathbf{w}_2} = \Omega_{\mathbf{w}_1} \times \Omega_{\mathbf{w}_2}$. To see that both $\Omega_{\mathbf{w}_1}$ and $\Omega_{\mathbf{w}_2}$ are convex sets, we check that $\forall\, \mathbf{w}_1^0, \mathbf{w}_1^1 \in \Omega_{\mathbf{w}_1}$ and $\forall\, \mathbf{w}_2^0, \mathbf{w}_2^1 \in \Omega_{\mathbf{w}_2}$

$$\mathcal{L}\mathbf{w}_1^a + \mathbf{J}_{p_1} = a(\mathcal{L}\mathbf{w}_1^0 + \mathbf{J}_{p_1}) + (1-a)(\mathcal{L}\mathbf{w}_1^1 + \mathbf{J}_{p_1}) \in \mathbb{S}^{p_1}_{++},\ \forall\, 0 < a < 1 \tag{27}$$

$$\mathcal{L}\mathbf{w}_2^b + \mathbf{J}_{p_2} = b(\mathcal{L}\mathbf{w}_2^0 + \mathbf{J}_{p_2}) + (1-b)(\mathcal{L}\mathbf{w}_2^1 + \mathbf{J}_{p_2}) \in \mathbb{S}^{p_2}_{++},\ \forall\, 0 < b < 1, \tag{28}$$

where $\mathbf{w}_1^a = a\mathbf{w}_1^0 + (1-a)\mathbf{w}_1^1 > \mathbf{0}$ and $\mathbf{w}_2^b = b\mathbf{w}_2^0 + (1-b)\mathbf{w}_2^1 > \mathbf{0}$. The set of positive definite matrices forms a convex cone. Now to prove $Q(\mathbf{w}_1, \mathbf{w}_2)$ is bi-convex, we again first consider the case where $\alpha_1 = \alpha_2 = 0$. The negative log-likelihood (15) is convex with respect to $\mathbf{w}$. Because either $\mathbf{w}_1$ or $\mathbf{w}_2$ maps linearly to $\mathbf{w}$ when the other factor is fixed, (15) is bi-convex. The penalized MLE is then also bi-convex because both $\alpha_1 \|\mathbf{w}_1\|_1$ and $\alpha_2 \|\mathbf{w}_2\|_1$ are convex. Thus each sub-problem of the penalized MLE has a unique solution. □

A.3 Proof of Corollary 4.3

Proof. We prove the first half of the corollary, then obtain the other half by symmetry. Let $\mathbf{L}^*$ be the Laplacian of the true Kronecker product graph to be estimated and $\mathcal{L}\mathbf{w}^* = \mathbf{L}^*$. By the properties of Kronecker graph product, $\mathbf{L}^* = \mathbf{D}^* - \mathbf{W}^* = \mathbf{D}_1^* \otimes \mathbf{D}_2^* - \mathbf{W}_1^* \otimes \mathbf{W}_2^*$. Let $\mathbf{L}_1^*$ and $\mathbf{L}_2^*$ be the true factor Laplacian, where $\mathcal{L}\mathbf{w}_1^* = \mathbf{L}_1^*$ and $\mathcal{L}\mathbf{w}_2^* = \mathbf{L}_2^*$. Although the factor Laplacians do not appear in the original problem formulation, they come in handy for deriving the consistency results.

Let's define a set of perturbations around $\mathbf{L}_1^*$

$$\mathcal{T}_1 = \{\Delta_{\mathbf{L}_1} | \Delta_{\mathbf{L}_1} \in \mathcal{K}_{\mathbf{L}^*}, \|\Delta_{\mathbf{L}_1}\|_F = c_1 r_{n,\mathbf{p}}\}, \tag{29}$$

where $r_{n,\mathbf{p}} = \sqrt{\frac{s_1 p_1 \log p}{n p_2}}$ for $\mathbf{p} = (p, p_1, p_2)$ and

$$\mathcal{K}_{\mathbf{L}_1^*} := \{\Delta_{\mathbf{L}_1} | \mathbf{L}_1^* + \Delta_{\mathbf{L}_1} \in \Omega_{\mathbf{L}_1}\}. \tag{30}$$

If we can show that

$$F(\Delta_{\mathbf{L}_1}) = Q(\mathbf{L}_1^* + \Delta_{\mathbf{L}_1}, \mathbf{L}_2^{\text{init}}) - Q(\mathbf{L}_1^*, \mathbf{L}_2^{\text{init}}) > 0,\ \forall \Delta_{\mathbf{L}_1} \in \mathcal{T}_1, \tag{31}$$

then we will have

$$\|\widehat{\mathbf{L}}_1 - \mathbf{L}_1^*\|_F \le c r_{n,\mathbf{p}}, \tag{32}$$

following that $Q(\mathbf{L}_1, \mathbf{L}_2)$ is bi-convex, $F(\mathbf{O}_1) = 0$, and $F(\widehat{\mathbf{L}}_1 - \mathbf{L}_1^*) = Q(\widehat{\mathbf{L}}_1, \mathbf{L}_2^{\text{init}}) - Q(\mathbf{L}_1^*, \mathbf{L}_2^{\text{init}}) \le 0$.

Let $\Delta_{\mathbf{L}} = \Delta_{\mathbf{D}_1} \otimes \mathbf{D}_2^{\text{init}} - \Delta_{\mathbf{W}_1} \otimes \mathbf{W}_2^{\text{init}}$. Then we can write

$$F(\Delta_{\mathbf{L}_1}) = \langle \Delta_{\mathbf{L}}, \mathbf{S} \rangle - (\log\det(\mathbf{L}^* + \Delta_{\mathbf{L}} + \mathbf{J}_p) - \log\det(\mathbf{L}^* + \mathbf{J}_p)) + \alpha_1(\|\mathbf{w}_1^* + \Delta_{\mathbf{w}_1}\|_1 - \|\mathbf{w}_1^*\|_1). \tag{33}$$

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

$$\begin{aligned} &\log\det(\mathbf{L}^* + \Delta_{\mathbf{L}} + \mathbf{J}_p) - \log\det(\mathbf{L}^* + \mathbf{J}_p) \\ =& \operatorname{Tr}((\mathbf{L}^* + \mathbf{J}_p)^{-1}\Delta_{\mathbf{L}}) + \int_0^1 (1-\nu)\nabla_\nu^2 \log\det(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p) d\nu, \end{aligned} \tag{34}$$

and further the remainder

$$\begin{aligned} &\int_0^1 (1-\nu)\nabla_\nu^2 \log\det(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p) d\nu \\ =& -\operatorname{vec}(\Delta_{\mathbf{L}})^T \left(\int_0^1 (1-\nu)(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} \otimes (\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} d\nu \right) \operatorname{vec}(\Delta_{\mathbf{L}}). \end{aligned} \tag{35}$$

Therefore we have

$$F(\Delta_{\mathbf{L}_1}) = I_1 + I_2 + I_3, \tag{36}$$

where

$$I_1 = \langle \Delta_{\mathbf{L}}, \mathbf{S} - (\mathbf{L}^* + \mathbf{J}_p)^{-1} \rangle, \tag{37}$$

$$I_2 = \mathrm{vec}(\Delta_{\mathbf{L}})^T \left(\int_0^1 (1-\nu)(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} \otimes (\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} d\nu \right) \mathrm{vec}(\Delta_{\mathbf{L}}), \tag{38}$$

$$I_3 = \alpha_1(\|\mathbf{w}_1^* + \Delta_{\mathbf{w}_1}\|_1 - \|\mathbf{w}_1^*\|_1). \tag{39}$$

Bound I_1: The observations $\bar{\mathbf{x}}$ are the samples from the improper GMRF [49]

$$\bar{\mathbf{x}} = \mathbf{x} - \frac{1}{p}\mathbf{1}\mathbf{1}^T\mathbf{x}, \quad \mathbf{x} \sim \mathcal{N}(\mathbf{0}, (\mathbf{L}^* + \mathbf{J}_p)^{-1}). \tag{40}$$

Let $\mathbf{\Sigma} = (\mathbf{L}^* + \mathbf{J}_p)^{-1}$ be the covariance matrix of the original proper GMRF. Let $m_1 = i - j + \frac{1}{2}(j-1)(2p_1 - j)$, $\forall 1 \le j < i \le p$ and $\mathcal{L}\Delta_{\mathbf{w}} = \Delta_{\mathbf{L}}$, we have

$$\begin{aligned}
I_1 &= \Delta_{\mathbf{w}}^T \mathcal{L}^*(\mathbf{S} - (\mathbf{L}^* + \mathbf{J}_p)^{-1}) && (41)\\
&= \frac{1}{2}\langle \Delta_{\mathbf{W}_1} \otimes \mathbf{W}_2^{\text{init}}, \mathcal{A}\mathcal{L}^*(\mathbf{S} - (\mathbf{L}^* + \mathbf{J}_p)^{-1}) \rangle && (42)\\
&= \sum_{1 \le j < i \le p_1} [\Delta_{\mathbf{w}_1}]_{m_1} \langle \mathcal{A}\mathbf{e}_{m_1} \otimes \mathbf{W}_2^{\text{init}}, \mathcal{A}\mathcal{L}^*(\mathbf{S} - \mathbf{\Sigma}) \rangle && (43)\\
&= \sum_{1 \le j < i \le p_1} [\Delta_{\mathbf{w}_1}]_{m_1} \langle \mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1} - [\mathcal{A}\mathcal{L}^*\mathbf{\Sigma}]_{I_1,J_1} \rangle && (44)\\
&= \sum_{1 \le j < i \le p_1} [\Delta_{\mathbf{w}_1}]_{m_1} \left(\langle \mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1} \rangle - \langle \mathbf{W}_2^{\text{init}}, [\mathcal{A}\mathcal{L}^*\mathbf{\Sigma}]_{I_1,J_1} \rangle \right), && (45)
\end{aligned}$$

where $\mathbf{e}_{m_1} \in \mathbb{R}^{\frac{p_1(p_1-1)}{2}}$ has 1 in the m_1-th entry and 0s otherwise. Also, notice that

$$\begin{aligned}
\mathbb{E}[[\mathcal{L}^*\mathbf{S}]_m] &= \mathbb{E}\left[\frac{1}{n} \sum_{k=1}^n ([\bar{\mathbf{x}_k}]_i - [\bar{\mathbf{x}_k}]_j)^2 \right] && (46)\\
&= \frac{1}{n} \sum_{k=1}^n \mathbb{E}\left[([\mathbf{x}_k]_i - [\mathbf{x}_k]_j)^2 \right] && (47)\\
&= \frac{1}{n} \sum_{k=1}^n \mathbb{E}[[\mathbf{x}_k]_i^2] - 2\mathbb{E}[[\mathbf{x}_k]_i[\mathbf{x}_k]_j] + \mathbb{E}[[\mathbf{x}_k]_j^2] && (48)\\
&= [\mathbf{\Sigma}]_{i,i} - [\mathbf{\Sigma}]_{i,j} - [\mathbf{\Sigma}]_{j,i} + [\mathbf{\Sigma}]_{j,j} && (49)\\
&= [\mathcal{L}^*\mathbf{\Sigma}]_m, && (50)
\end{aligned}$$

which leads to

$$\mathbb{E}[\langle \mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1} \rangle] = \langle \mathbf{W}_2^{\text{init}}, \mathbb{E}[[\mathbf{K}]_{I_1,J_1}] \rangle = \langle \mathbf{W}_2^{\text{init}}, [\mathcal{A}\mathcal{L}^*\mathbf{\Sigma}]_{I_1,J_1} \rangle. \tag{51}$$

Therefore,

$$I_1 = \sum_{1 \le j < i \le p_1} [\Delta_{\mathbf{w}_1}]_{m_1} \left(\langle \mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1} \rangle - \mathbb{E}[\langle \mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1} \rangle] \right). \tag{52}$$

Now we bound the perturbation term. Let $\mathbf{x}_k = \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{z}_k$, where $\mathbf{z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$ is the source signal of the GSP system. From (43) we know

$$\begin{aligned}
\langle \mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1} \rangle &= \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k^T (\mathcal{L}\mathcal{A}^*(\mathcal{A}\mathbf{e}_{m_1} \otimes \mathbf{W}_2^{\text{init}}))\mathbf{x}_k && (53)\\
&= \frac{1}{n} \sum_{k=1}^n \mathbf{z}_k^T \mathbf{\Sigma}^{\frac{1}{2}} (\mathcal{L}\mathcal{A}^*(\mathcal{A}\mathbf{e}_{m_1} \otimes \mathbf{W}_2^{\text{init}}))\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{z}_k. && (54)
\end{aligned}$$

Let $\mathbf{M}_{i,j} = \mathbf{\Sigma}^{\frac{1}{2}}(\mathcal{L}\mathcal{A}^*(\mathcal{A}\mathbf{e}_{m_1} \otimes \mathbf{W}_2^{\text{init}}))\mathbf{\Sigma}^{\frac{1}{2}}$. We then apply the Hanson-Wright inequality [50, 51] to the quadratic

$$\mathbb{P}\left\{\left|\frac{1}{n}\sum_{k=1}^{n}\mathbf{z}_k^T\mathbf{M}_{i,j}\mathbf{z}_k - \mathbb{E}\left[\frac{1}{n}\sum_{k=1}^{n}\mathbf{z}_k^T\mathbf{M}_{i,j}\mathbf{z}_k\right]\right| > h\right\} \tag{55}$$

$$\leq 2\exp\left[-c_1'\min\left(\frac{nh^2}{K^4\|\mathbf{M}_{i,j}\|_F^2}, \frac{nh}{K^2\|\mathbf{M}_{i,j}\|_2}\right)\right] \tag{56}$$

$$\leq 2\exp\left[-c_1'\min\left(\frac{nh^2}{K^4\|\mathbf{M}_{i,j}\|_F^2}, \frac{nh}{K^2\|\mathbf{M}_{i,j}\|_F}\right)\right] \tag{57}$$

$$\leq 2\exp\left[-c_1'\min\left(\frac{nh^2}{2K^4\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F^2\|\mathbf{\Sigma}\|_2^2}, \frac{nh}{\sqrt{2}K^2\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F\|\mathbf{\Sigma}\|_2}\right)\right] \tag{58}$$

$$\leq 2\exp\left[-c_1'\min\left(\frac{nh^2}{32\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F^2\|\mathbf{\Sigma}\|_2^2}, \frac{nh}{4\sqrt{2}\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F\|\mathbf{\Sigma}\|_2}\right)\right], \tag{59}$$

where from (57) to (58) we use

$$\|\mathbf{M}_{i,j}\|_F^2 \leq \|\mathbf{\Sigma}^{\frac{1}{2}}\|_2^2\|\mathcal{L}\mathcal{A}^*(\mathcal{A}\mathbf{e}_{m_1} \otimes \mathbf{W}_2^{\text{init}})\|_F^2\|\mathbf{\Sigma}^{\frac{1}{2}}\|_2^2 = 2\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F^2\|\mathbf{\Sigma}\|_2^2, \tag{60}$$

and from (58) to (59) we use $K \leq 2$ for the sub-Gaussian norm of $\mathbf{z}_k$. Let $\epsilon = \frac{h}{\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F\|\mathbf{\Sigma}\|_2}$ and plug (54) into (59), we arrive at

$$\mathbb{P}\left\{|\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle - \mathbb{E}[\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle]| > \epsilon\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F\|\mathbf{\Sigma}\|_2\right\} \leq 2\exp\left(-\frac{c_1'n\epsilon^2}{32}\right), \ \forall\epsilon \leq 4\sqrt{2}. \tag{61}$$

The union bound indicates that

$$\mathbb{P}\left\{\max_{m_1}\left[|\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle - \mathbb{E}[\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle]|\right] > \epsilon\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F\|\mathbf{\Sigma}\|_2\right\} \tag{62}$$

$$\leq \sum_{m_1=1}^{\frac{p_1(p_1-1)}{2}} 2\exp\left(-\frac{c_1'n\epsilon^2}{32}\right) \tag{63}$$

$$\leq p_1^2\exp\left(-\frac{c_1'n\epsilon^2}{32}\right), \tag{64}$$

so

$$\mathbb{P}\left\{\max_{m_1}\left[|\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle - \mathbb{E}[\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle]|\right] \leq \epsilon\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F\|\mathbf{\Sigma}\|_2\right\} \geq 1 - p_1^2\exp\left(-\frac{c_1'n\epsilon^2}{32}\right). \tag{65}$$

Thus with the above probability and $\epsilon \leq 4\sqrt{2}$

$$I_1 = \sum_{1\leq j<i\leq p_1}[\Delta_{\mathbf{w}_1}]_{m_1}\left(\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle - \mathbb{E}[\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle]\right) \tag{66}$$

$$\geq -\max_{m_1}\left[|\langle\mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1} - [\mathcal{A}\mathcal{L}^*\mathbf{\Sigma}]_{I_1,J_1}\rangle|\right]\|\Delta_{\mathbf{w}_1}\|_1 \tag{67}$$

$$\geq -\epsilon\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F\|\mathbf{\Sigma}\|_2\|\Delta_{\mathbf{w}_1}\|_1 \tag{68}$$

$$\geq -\epsilon\sqrt{p_2}\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2\|\Delta_{\mathbf{w}_1}\|_1. \tag{69}$$

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

$$I_2 \geq \|\Delta_{\mathbf{L}}\|_F^2\lambda_{\min}\left(\int_0^1(1-\nu)(\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1} \otimes (\mathbf{L}^* + \nu\Delta_{\mathbf{L}} + \mathbf{J}_p)^{-1}d\nu\right). \tag{70}$$

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

$$\lambda_{\min}\left(\int_0^1 (1-\nu)(\mathbf{L}^*+\nu\Delta_{\mathbf{L}}+\mathbf{J}_p)^{-1}\otimes(\mathbf{L}^*+\nu\Delta_{\mathbf{L}}+\mathbf{J}_p)^{-1}d\nu\right) \tag{71}$$

$$\geq \int_0^1 (1-\nu)\lambda_{\min}^2(\mathbf{L}^*+\nu\Delta_{\mathbf{L}}+\mathbf{J}_p)^{-1}d\nu \tag{72}$$

$$\geq \min_{\nu\in[0,1]}\left[\lambda_{\min}^2(\mathbf{L}^*+\nu\Delta_{\mathbf{L}}+\mathbf{J}_p)^{-1}\right]\int_0^1(1-\nu)d\nu \tag{73}$$

$$=\frac{1}{2}\min_{\nu\in[0,1]}\left[\frac{1}{\|\mathbf{L}^*+\nu\Delta_{\mathbf{L}}+\mathbf{J}_p\|_2}^2\right] \tag{74}$$

$$=\frac{1}{2\max_{\nu\in[0,1]}\left[\|\mathbf{L}^*+\nu\Delta_{\mathbf{L}}+\mathbf{J}_p\|_2^2\right]} \tag{75}$$

$$\geq\frac{1}{2\max^2_{\nu\in[0,1]}\left[\|\mathbf{L}^*+\mathbf{J}_p\|_2+\|\nu\Delta_{\mathbf{L}}\|_2\right]} \tag{76}$$

$$=\frac{1}{2(\|\mathbf{L}^*+\mathbf{J}_p\|_2+\|\Delta_{\mathbf{L}}\|_2)^2} \tag{77}$$

Let $\mathbf{d}_1$ denotes the diagonal of $\mathbf{D}_1$. The Gershgorin circle theorem implies that

$$\|\Delta_{\mathbf{L}}\|_2 \leq 2\|\Delta_{\mathbf{d}_1}\|_\infty d_{2,\max} \leq 2d_{2,\max}\|\Delta_{\mathbf{L}_1}\|_F = 2cd_{2,\max}r_{n,\mathbf{p}} \tag{78}$$

Then with n sufficiently large $n \geq \frac{4c^2d_{2,\max}^2 s_1 p_1 \log p}{p_2\|\mathbf{L}^*+\mathbf{J}_p\|_2^2}$ such that $\|\Delta_{\mathbf{L}}\|_2 \leq 2cd_{2,\max}r_{n,\mathbf{p}} \leq \|\mathbf{L}^*+\mathbf{J}_p\|_2$, we obtain

$$I_2 \geq \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\|\mathbf{L}^*+\mathbf{J}_p\|_2^2}. \tag{79}$$

To factor $\Delta_{\mathbf{L}_1}$ out, note that

$$\|\Delta_{\mathbf{L}}\|_F^2 = \|\Delta_{\mathbf{W}_1}\otimes\mathbf{W}_2^{\text{init}}\|_F^2 + \|\Delta_{\mathbf{D}_1}\otimes\mathbf{D}_2^{\text{init}}\|_F^2 \tag{80}$$

$$= \|\Delta_{\mathbf{W}_1}\|_F^2\|\mathbf{W}_2^{\text{init}}\|_F^2 + \|\Delta_{\mathbf{D}_1}\|_F^2\|\mathbf{D}_2^{\text{init}}\|_F^2 \tag{81}$$

$$\geq \|\Delta_{\mathbf{L}_1}\|_F^2\|\mathbf{W}_2^{\text{init}}\|_F^2, \tag{82}$$

and that $\|\mathbf{W}_2^{\text{init}}\|$ is lower and upper bounded

$$\sqrt{\frac{p_2}{s_2}}d_{2,\min} \leq \|\mathbf{W}_2^{\text{init}}\|_F \leq \sqrt{p_2}d_{2,\max} \tag{83}$$

Thus

$$I_2 \geq \frac{p_2 d_{2,\min}^2\|\Delta_{\mathbf{L}_1}\|_F^2}{8s_2\|\mathbf{L}^*+\mathbf{J}_p\|_2^2}. \tag{84}$$

Bound I_3: With the triangle inequality

$$\|\mathbf{w}_1^*+\Delta_{\mathbf{w}_1}\|_1 - \|\mathbf{w}_1^*\|_1 = \|\mathbf{w}_1^*+\Delta_{\mathbf{w}_1}\|_{1,\mathcal{S}_1} + \|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{S}_1^\complement} - \|\mathbf{w}_1^*\|_{1,\mathcal{S}_1} \geq \|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{S}_1^\complement} - \|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{S}_1}, \tag{85}$$

we can lower-bound I_3

$$I_3 \geq 2\alpha_1\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{S}_1^\complement} - 2\alpha_1\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{S}_1}. \tag{86}$$

Bound $I_1+I_2+I_3$: Overall,

$$F(\Delta_{\mathbf{L}_1}) \geq -\epsilon\sqrt{p_2}\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2\|\Delta_{\mathbf{w}_1}\|_1 + \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\|\mathbf{L}^*+\mathbf{J}_p\|_2^2} + 2\alpha_1\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{A}_1^\complement} - 2\alpha_1\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{A}_1} \tag{87}$$

$$=\frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\|\mathbf{L}^*+\mathbf{J}_p\|_2^2} - (\epsilon\sqrt{p_2}\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2 - 2\alpha_1)\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{A}_1^\complement} \tag{88}$$

$$- (\epsilon\sqrt{p_2}\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2 + 2\alpha_1)\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{A}_1}.$$

Let $\epsilon = c_1'' \sqrt{\frac{\log p}{n}}$ with sufficiently large

$$n \geq \frac{c_1''^2 \log p}{32}, \tag{89}$$

so that $\epsilon \leq 4\sqrt{2}$ is satisfied. Then choose

$$\alpha_1 \geq \frac{c_1'' \|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2 \|\mathbf{\Sigma}\|_2}{2} \sqrt{\frac{p_2 \log p}{n}}, \tag{90}$$

so that $\epsilon\sqrt{p_2}\|\mathcal{L}\mathbf{w}_2\|_2\|\mathbf{\Sigma}\|_2 - 2\alpha < 0$ and

$$F(\Delta_{\mathbf{L}_1}) \geq \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\|\mathbf{L}^* + \mathbf{J}_p\|_2^2} - (\epsilon\sqrt{p_2}\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2 + 2\alpha_1)\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{A}_1}, \tag{91}$$

We can also bound $\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{A}_1}$ by

$$\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{S}_1} \leq \sqrt{p_1 s_1}\|\Delta_{\mathbf{w}_1}\|_{2,\mathcal{S}_1} \leq \sqrt{p_1 s_1}\|\Delta_{\mathbf{w}_1}\|_2 \leq \sqrt{\frac{p_1 s_1}{2}}\|\Delta_{\mathbf{L}_1}\|_F. \tag{92}$$

Now, to prove $F(\Delta_{\mathbf{L}_1}) \geq 0$, define a ratio factor $\gamma_1 \geq 1$ that controls the ℓ_1 penalty $\frac{\alpha_1}{\gamma_1} = \frac{c_1''\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2}{2}\sqrt{\frac{p_2 \log p}{n}}$. We obtain

$$F(\Delta_{\mathbf{L}_1}) \geq \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\|\mathbf{L}^* + \mathbf{J}_p\|_2^2} - (1+\gamma_1)c_1''\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2\sqrt{\frac{p_2 \log p}{n}}\|\Delta_{\mathbf{w}_1}\|_{1,\mathcal{S}_1} \tag{93}$$

$$\geq p_2\|\Delta_{\mathbf{L}_1}\|_F^2 \left(\frac{d_{2,\min}^2}{8s_2\|\mathbf{L}^* + \mathbf{J}_p\|_2^2} - (1+\gamma_1)c_1''\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2\sqrt{\frac{s_1 p_1 \log p}{2p_2 n}}\|\Delta_{\mathbf{L}_1}\|_F^{-1}\right) \tag{94}$$

$$= p_2\|\Delta_{\mathbf{L}_1}\|_F^2 \left(\frac{d_{2,\min}^2}{8s_2\|\mathbf{L}^* + \mathbf{J}_p\|_2^2} - (1+\gamma_1)\frac{c_1''\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2}{\sqrt{2}c_1}\right) \tag{95}$$

$$> 0, \tag{96}$$

for sufficiently large c

$$c_1 \geq 4\sqrt{2}(1+\gamma_1)\frac{c_1'' s_2}{d_{2,\min}^2}\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_2\|\mathbf{\Sigma}\|_2\|\mathbf{L}^* + \mathbf{J}_p\|_2^2. \tag{97}$$

This happens with probability

$$\mathbb{P}\left\{\max_{m_1}\left[|\langle \mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle - \mathbb{E}[\langle \mathbf{W}_2^{\text{init}}, [\mathbf{K}]_{I_1,J_1}\rangle]|\right] \leq \epsilon\|\mathcal{L}\mathbf{w}_2^{\text{init}}\|_F\|\mathbf{\Sigma}\|_2\right\} \tag{98}$$

$$\geq 1 - p_1^2 \exp\left(-\frac{c_1' n\epsilon^2}{32}\right) \tag{99}$$

$$= 1 - \exp\left[2\log p_1 - \frac{c_1' c_1''^2}{32}\log p\right] \tag{100}$$

$$\geq 1 - \exp\left(-\left(\frac{c_1' c_1''^2}{32} - 2\right)\log p\right). \tag{101}$$

We have proved that the $\mathbf{w}_1$ estimation is consistent when fixing $\mathbf{w}_2$. □

A.4 Proof of Theorem 4.4

Proof. The algorithm starts with fixing $\mathbf{w}_2 = \mathbf{w}_2^{\text{init}}$ and updating $\mathbf{w}_1$, whose error is obtained by Corollary 4.3. Now we move forward to prove the consistency of $\mathbf{w}_2$ when fixing $\widehat{\mathbf{w}}_1 = \mathbf{w}_1^{(1)}$. Similarly, we aim to show

$$G(\Delta_{\mathbf{L}_2}) = Q(\mathbf{L}_1^{(1)}, \mathbf{L}_2^* + \Delta_{\mathbf{L}_2}) - Q(\mathbf{L}_1^{(1)}, \mathbf{L}_2^*) > 0, \ \forall \Delta_{\mathbf{L}_2} \in \mathcal{T}_2, \tag{102}$$

$$\mathcal{T}_2 = \{\Delta_{\mathbf{L}_2} | \Delta_{\mathbf{L}_2} \in \mathcal{K}_{\mathbf{L}^*}, \|\Delta_{\mathbf{L}_2}\|_F = c_2 r_{n,\mathbf{p}}\}, \tag{103}$$

where $r_{n,\mathbf{p}} = \sqrt{\frac{s_2 p_2 \log p}{n p_1}}$. By symmetry, with high probability, for $\epsilon = c_2'' \sqrt{\frac{\log p}{n}}$ and $n \geq \frac{c_2''^2 \log p}{32}$

$$G(\Delta_{\mathbf{L}_2}) \geq -\epsilon\sqrt{p_1}\|\mathcal{L}\mathbf{w}_1^{(1))}\|_2\|\mathbf{\Sigma}\|_2\|\Delta_{\mathbf{w}_2}\| + \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\|\mathbf{L}^* + \mathbf{J}_p\|_2^2} + 2\alpha_2\|\Delta_{\mathbf{w}_2}\|_{1,\mathcal{A}_2^{\complement}} - 2\alpha_2\|\Delta_{\mathbf{w}_2}\|_{1,\mathcal{A}_2} \tag{104}$$

$$\begin{aligned} =& \frac{\|\Delta_{\mathbf{L}}\|_F^2}{8\|\mathbf{L}^* + \mathbf{J}_p\|_2^2} - (\epsilon\sqrt{p_1}\|\mathcal{L}\mathbf{w}_1^{(1)}\|_2\|\mathbf{\Sigma}\|_2 - 2\alpha_2)\|\Delta_{\mathbf{w}_2}\|_{1,\mathcal{A}_2^{\complement}} \\ &- (\epsilon\sqrt{p_1}\|\mathcal{L}\mathbf{w}_1^{(1)}\|_2\|\mathbf{\Sigma}\|_2 + 2\alpha_2)\|\Delta_{\mathbf{w}_2}\|_{1,\mathcal{A}_2}, \end{aligned} \tag{105}$$

where $\Delta_{\mathbf{L}} = \mathbf{D}_1^{(1)} \otimes \Delta_{\mathbf{D}_2} - \mathbf{W}_1^{(1)} \otimes \Delta_{\mathbf{W}_2}$. Different from the previous derivation, $\mathbf{L}_1^{(1)}$ and $\mathbf{W}_1^{(1)}$ are now variables and we use $\|\mathbf{L}_1^{(1)} - \mathbf{L}_1^*\|_F \leq c\sqrt{\frac{s_1 p_1 \log p}{n p_2}}$ to bound them. First, we have

$$\|\mathbf{W}_1^{(1)}\|_F^2 \geq \|\mathbf{W}_1^*\|_F^2 - 2\|\mathbf{W}_1^*\|_F\|\mathbf{W}_1^{(1)} - \mathbf{W}_1^*\|_F \geq \frac{d_{1,\min}^2 p_1}{s_1} - 2c_1 d_{1,\max} p_1 \sqrt{\frac{s_1 \log p}{n p_2}}. \tag{106}$$

Then, let $d_{1,\max}^{(1)}$ be the maximum degree of $\mathbf{W}_1^{(1)}$, by the Gershgorin circle theorem

$$\|\mathbf{L}_1^{(1)}\|_2 \leq 2d_{1,\max}^{(1)} \leq 2d_{1,\max} + 2c_1\sqrt{\frac{s_1 p_1 \log p}{n p_2}}. \tag{107}$$

Similar to (90), we choose a large enough α_2 with a ratio factor $\gamma_2 \geq 1$

$$\frac{\alpha_2}{\gamma_2} = c_2''\left(d_{1,\max} + c_1\sqrt{\frac{s_1 p_1 \log p}{n p_2}}\right)\|\mathbf{\Sigma}\|_2\sqrt{\frac{p_1 \log p}{n}}. \tag{108}$$

Similar to (82) and (92), we have

$$\|\Delta_{\mathbf{L}}\|_F^2 \geq \|\Delta_{\mathbf{L}_2}\|_F^2\|\mathbf{W}_1^{(1))}\|_F^2, \tag{109}$$

and

$$\|\Delta_{\mathbf{w}_2}\|_{1,\mathcal{S}_2} \leq \sqrt{p_2 s_2}\|\Delta_{\mathbf{w}_2}\|_{2,\mathcal{S}_2} \leq \sqrt{p_2 s_2}\|\Delta_{\mathbf{w}_2}\|_2 \leq \sqrt{\frac{p_2 s_2}{2}}\|\Delta_{\mathbf{L}_2}\|_F. \tag{110}$$

Plugging in $\epsilon = c_2''\sqrt{\frac{\log p}{n}}$, (108), (109), and (110), we obtain

$$G(\Delta_{\mathbf{L}_2}) \geq \|\Delta_{\mathbf{L}_2}\|_F^2\left(\frac{\|\mathbf{W}_1^{(1)}\|_F^2}{8\|\mathbf{L}^* + \mathbf{J}_p\|_2^2} - (1+\gamma_2)c_2''\sqrt{\frac{s_2 p_2 \log p}{2 n p_1}}\|\mathcal{L}\mathbf{w}_1^{(1))}\|_2\|\mathbf{\Sigma}\|_2\|\Delta_{\mathbf{L}_2}\|_F^{-1}\right) \tag{111}$$

$$\begin{aligned} \geq& p_1\|\Delta_{\mathbf{L}_2}\|_F^2\left(\frac{1}{8\|\mathbf{L}^* + \mathbf{J}_p\|_2^2}\left(\frac{d_{1,\min}^2}{s_1} - 2c_1 d_{1,\max}\sqrt{\frac{s_1 \log p}{n p_2}}\right)\right. \\ &\left. -\sqrt{2}(1+\gamma_2)\frac{c_2''}{c_2}\left(d_{1,\max} + c_1\sqrt{\frac{s_1 p_1 \log p}{n p_2}}\right)\|\mathbf{\Sigma}\|_2\right) \end{aligned} \tag{112}$$

$$\geq p_1\|\Delta_{\mathbf{L}_2}\|_F^2\left(\frac{(1-\zeta)d_{1,\min}^2}{8 s_1\|\mathbf{L}^* + \mathbf{J}_p\|_2^2} - \sqrt{2}(1+\gamma_2)(1+\iota)\frac{c_2'' d_{1,\max}}{c_2}\|\mathbf{\Sigma}\|_2\right) \tag{113}$$

$$\geq 0, \tag{114}$$

for sufficiently large n

$$n \geq \max\left[\frac{2c_1 d_{1,\max} s_1^2 \log p}{\zeta^2 d_{1,\min}^4 p_2}, \frac{c_1^2 s_1 p_1 \log p}{\iota^2 d_{1,\max}^2 p_2}\right], \tag{115}$$

and c_2

$$c_2 \geq 8\sqrt{2}\frac{(1+\gamma)(1+\iota)c_2'' d_{1,\max} s_1}{(1-\zeta)d_{1,\min}^2}\|\mathbf{\Sigma}\|_2 \|\mathbf{L}^* + \mathbf{J}_p\|_2^2, \tag{116}$$

where $0 < \zeta < 1$ and $0 < \iota$ are additional ratio factors. We have now proved that the second iteration of the alternating optimization is consistent. With induction, one can show that

$$\|\mathbf{L}_1^{(2t-1)} - \mathbf{L}_1^*\|_F \leq c_1\sqrt{\frac{s_1 p_1 \log p}{n p_2}}, \tag{117}$$

$$\|\mathbf{L}_2^{(2t)} - \mathbf{L}_2^*\|_F \leq c_2\sqrt{\frac{s_2 p_2 \log p}{n p_1}}. \tag{118}$$

Finally, to show the convergence of the product graphs, we decompose the error as

$$\begin{aligned}
\Delta_{\mathbf{W}^{(2t)}} &= \mathbf{W}_1^{(2t-1)} \otimes \mathbf{W}_2^{(2t)} - \mathbf{W}_1^* \otimes \mathbf{W}_2^* && (119)\\
&= \mathbf{W}_1^* \otimes \Delta_{\mathbf{W}_2^{(2t)}} + \Delta_{\mathbf{W}_1^{(2t-1)}} \otimes \mathbf{W}_2^* + \Delta_{\mathbf{W}_1^{(2t-1)}} \otimes \Delta_{\mathbf{W}_2^{(2t)}}, && (120)
\end{aligned}$$

from which we obtain

$$\begin{aligned}
\|\Delta_{\mathbf{W}^{(2t)}}\|_F &\leq \|\mathbf{W}_1^* \otimes \Delta_{\mathbf{W}_2^{(2t)}}\|_F + \|\Delta_{\mathbf{W}_1^{(2t-1)}} \otimes \mathbf{W}_2^*\|_F + \|\Delta_{\mathbf{W}_1^{(2t-1)}} \otimes \Delta_{\mathbf{W}_2^{(2t)}}\|_F && (121)\\
&= \|\mathbf{W}_1^*\|_F \|\Delta_{\mathbf{W}_2^{(2t)}}\|_F + \|\Delta_{\mathbf{W}_1^{(2t-1)}}\|_F \|\mathbf{W}_2^*\|_F + \|\Delta_{\mathbf{W}_1^{(2t-1)}}\|_F \|\Delta_{\mathbf{W}_2^{(2t)}}\|_F && (122)\\
&\leq c_1 d_{1,\max}\sqrt{\frac{s_2 p_2 \log p}{2n}} + c_2 d_{2,\max}\sqrt{\frac{s_1 p_1 \log p}{2n}} + c_1 c_2 \sqrt{s_1 s_2}\frac{\log p}{2n} && (123)\\
&\leq \sqrt{s_1 s_2}\max\left[\frac{c_1 d_{1,\max}}{\sqrt{s_1}}, \frac{c_2 d_{2,\max}}{\sqrt{s_2}}\right](\sqrt{p_1} + \sqrt{p_2})\sqrt{\frac{\log p}{2n}} + c_1 c_2 \sqrt{s_1 s_2}\frac{\log p}{2n} && (124)\\
&\leq (1+\kappa)\sqrt{s_1 s_2}\max\left[\frac{c_1 d_{1,\max}}{\sqrt{s_1}}, \frac{c_2 d_{2,\max}}{\sqrt{s_2}}\right](\sqrt{p_1} + \sqrt{p_2})\sqrt{\frac{\log p}{2n}} && (125)\\
&\leq c\sqrt{\frac{(p_1+p_2)\log p}{n}}. && (126)
\end{aligned}$$

Here again for κ the ratio factor

$$n \geq \frac{c_1^2 c_2^2 \log p}{2\kappa^2 \max^2\left[\frac{c_1 d_{1,\max}}{\sqrt{s_1}}, \frac{c_2 d_{2,\max}}{\sqrt{s_2}}\right](\sqrt{p_1} + \sqrt{p_2})^2}, \tag{127}$$

and

$$c = (1+\kappa)\sqrt{s_1 s_2}\max\left[\frac{c_1 d_{1,\max}}{\sqrt{s_1}}, \frac{c_2 d_{2,\max}}{\sqrt{s_2}}\right]. \tag{128}$$

Since the above also holds for $\|\Delta_{\mathbf{W}^{(2t+1)}}\|_F$, we have proved

$$\|\Delta_{\mathbf{W}^{(t)}}\|_F \leq c\sqrt{\frac{(p_1+p_2)\log p}{n}},\ \forall\, t \geq 2, \tag{129}$$

with a high probability for sufficiently large n. □

B Competing Methods

PST is a GSP method that extracts the eigenvectors of factor GSOs from the signal covariance. In [31], the GSO is set to be the weighted adjacency matrix of the product graph, which is the Kronecker product of the weighted adjacency matrices of the factor graphs $\mathbf{A} = \mathbf{A}_1 \otimes \mathbf{A}_2$. In this case, the eigenvectors of the factor-wise covariance matrices converge to the eigenvectors of the factor-weighted adjacency matrices. PST uses these eigenvectors, i.e. spectral templates, as a proxy, and solves for the eigenvalues that render a sparse graph. Although this is generally not feasible when the GSO is Laplacian as $\mathbf{L} \neq \mathbf{L}_1 \otimes \mathbf{L}_2$, it has been shown that for $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}$, $\mathbf{L}_1 = \mathbf{U}_1\mathbf{\Lambda}_1\mathbf{U}_1$, and $\mathbf{L}_2 = \mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2$, the Kronecker product of factor Laplacian eigenvectors can approximate the eigenvectors of the product Laplacian $\mathbf{U} \approx \mathbf{U}_1 \otimes \mathbf{U}_2$ [52, 53]. Let $\boldsymbol{\psi} = \text{vec}(\boldsymbol{\Psi})$ be the spectral representation of the smooth graph signals, we have

$$\text{vec}(\boldsymbol{X}) = \boldsymbol{x} = \mathbf{U}\boldsymbol{\phi} \approx (\mathbf{U}_1 \otimes \mathbf{U}_2)\boldsymbol{\psi} = \text{vec}(\mathbf{U}_2\,\boldsymbol{\Psi}\mathbf{U}_1^T). \tag{130}$$

Therefore the factor-wise covariance

$$\mathbb{E}[\boldsymbol{X}\,\boldsymbol{X}^T] = \mathbb{E}[\mathbf{U}_2\,\boldsymbol{\Psi}\mathbf{U}_1^T\mathbf{U}_1\,\boldsymbol{\Psi}^T\mathbf{U}_2^T] = \mathbf{U}_2\mathbb{E}[\boldsymbol{\Psi}\,\boldsymbol{\Psi}^T]\mathbf{U}_2^T, \tag{131}$$

$$\mathbb{E}[\boldsymbol{X}^T\boldsymbol{X}] = \mathbb{E}[\mathbf{U}_1\,\boldsymbol{\Psi}^T\mathbf{U}_2^T\mathbf{U}_2\,\boldsymbol{\Psi}\mathbf{U}_1^T] = \mathbf{U}_1^T\mathbb{E}[\boldsymbol{\Psi}^T\,\boldsymbol{\Psi}]\mathbf{U}_1. \tag{132}$$

Following Theorem 2 in [31], one can show that $\mathbb{E}[\boldsymbol{\Psi}\,\boldsymbol{\Psi}^T]$ and $\mathbb{E}[\boldsymbol{\Psi}^T\,\boldsymbol{\Psi}]$ are both diagonal matrices. This means that the eigenvectors of the factor-wise covariance matrices approximate the eigenvectors of factor Laplacians.

FF is an alternating algorithm for estimating the Kronecker structured covariance matrices $\mathbf{\Sigma} = \mathbf{A} \otimes \mathbf{B}$ in matrix normal distributions. By the property of the Kronecker product, $\det(\mathbf{A}^{-1} \otimes \mathbf{B}^{-1}) = \det(\mathbf{A}^{-1})^{p_2}\det(\mathbf{B}^{-1})^{p_1}$. Therefore, the MLE of the matrix normal distribution simplifies to a concise form

$$\min_{\mathbf{A}\in\mathbb{S}_{++}^{p_1},\mathbf{B}\in\mathbb{S}_{++}^{p_2}} \left\{\langle \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}, \mathbf{S}\rangle - p_2 \log\det(\mathbf{A}^{-1}) - p_1 \log\det(\mathbf{B}^{-1})\right\}, \tag{133}$$

and FF alternates between closed-form updates of $\mathbf{A}$ and $\mathbf{B}$ to solve the problem. **KGLasso** adds ℓ-1 sparsity penalization to the problem and uses graphical lasso to solve each sub-problem. Note that this MLE differs from our Kronecker graph learning problem, because the Laplacian of the Kronecker product graph cannot factor $\mathbf{L} \neq \mathbf{L}_1 \otimes \mathbf{L}_2$. This poses more difficulties in solving our penalized MLE as closed-form solutions no longer exist. Therefore, using these GM methods to solve our problem causes a model mismatch. **PGL** [30], **TeraLasso** [45], and **MWGL** [32] are Cartesian graph learning methods. These methods are not appropriate for Kronecker product graph learning but serve as baselines for strong product graph learning. Please refer to their original papers for the details.

C Additional Simulation Results

Here we provide additional results from synthetic experiments. Fig. 5 shows the PR-AUC of Kronecker product graph learning and Fig. 6 shows the PR-AUC of strong product graph learning. KSGL outperforms all competing methods on both Kronecker and strong product graph learning. For Kronecker product graphs, only KSGL perfectly learns the true underlying graphs as the number of signals increases. For strong product graphs, KSGL requires fewer graph signals to fully reconstruct the edge pattern. Fig. 7 shows the results of applying Cartesian product methods TeraLasso and MWGL to Kronecker product graph learning. As we can see, these methods under-perform KSGL due to model mismatch as expected.

D Additional EEG Results

In Sec. 6.2, we apply KSGL to the EEG signals of a patient whose seizures are not visible on EEG. Although the signal amplitude does not manifest the ongoing seizures, the brain graph learned from epileptic signals shows increased connectivity compared with the normal one. In Fig. 8, we also show that KSGL learns more distinct temporal connectivity than MWGL, while the epileptic signals are more knitted than the normal signals for both methods. Additionally, we select another type of patient whose seizures are visible on the EEG. These patients all suffer from complex partial seizures, and we apply KSGL to their normal and epileptic EEG signals. Fig. 9 shows the node degree distributions of the learned brain graphs. We observe that KSGL learns denser connectivity from the normal EEG and sparser connectivity from the abnormal EEG, and this pattern is consistent across all four patients.

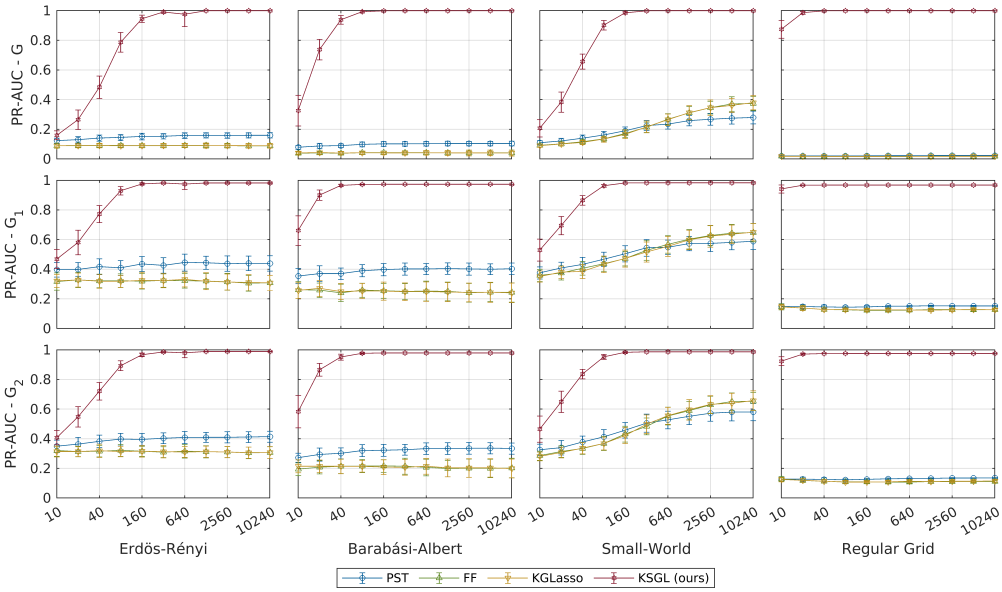


Figure 5: Comparison of different methods on various synthetic Kronecker product graphs and signals. Each sub-figure shows the trend of PR-AUC of the product or factor edge prediction as n increases.

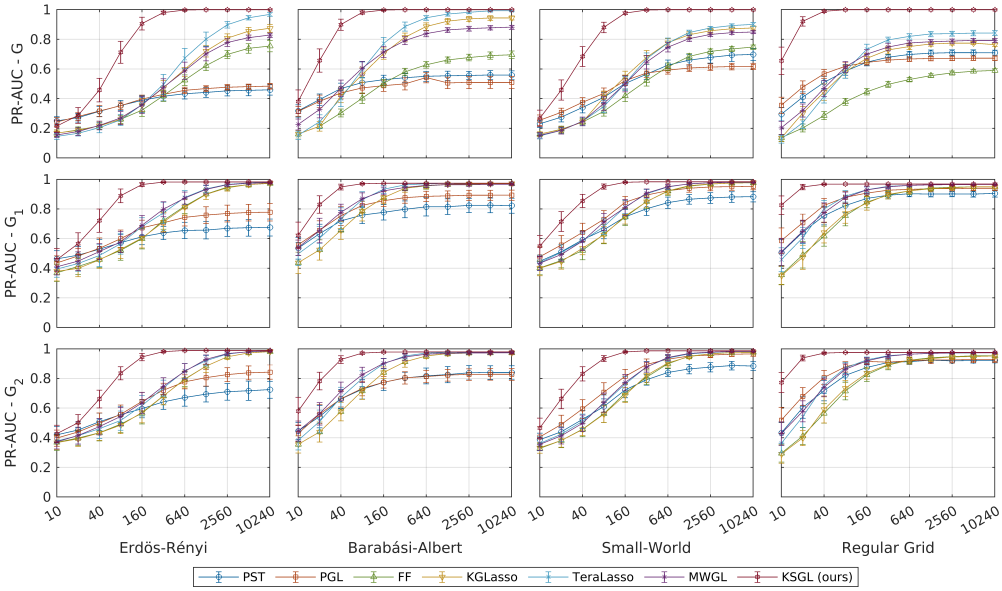


Figure 6: Comparison of different methods on various synthetic strong product graphs and signals. Each sub-figure shows the trend of PR-AUC of the product or factor edge prediction as n increases.

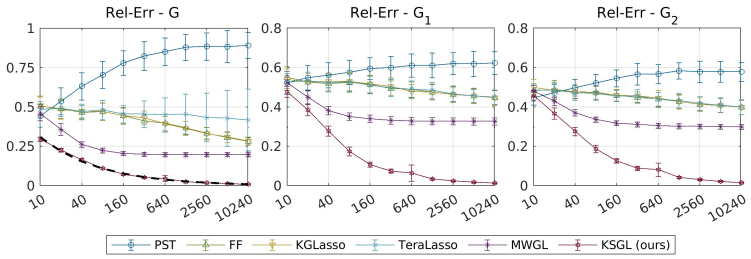
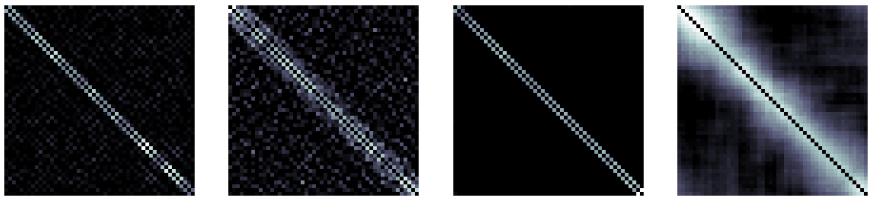


Figure 7: Applying Cartesian product graph learning methods to learn Kronecker product graphs.



(a) MWGL - normal **(b)** MWGL - seizure **(c)** KSGL - normal **(d)** KSGL - seizure

Figure 8: The time adjacency matrices inferred by MWGL and KSGL.

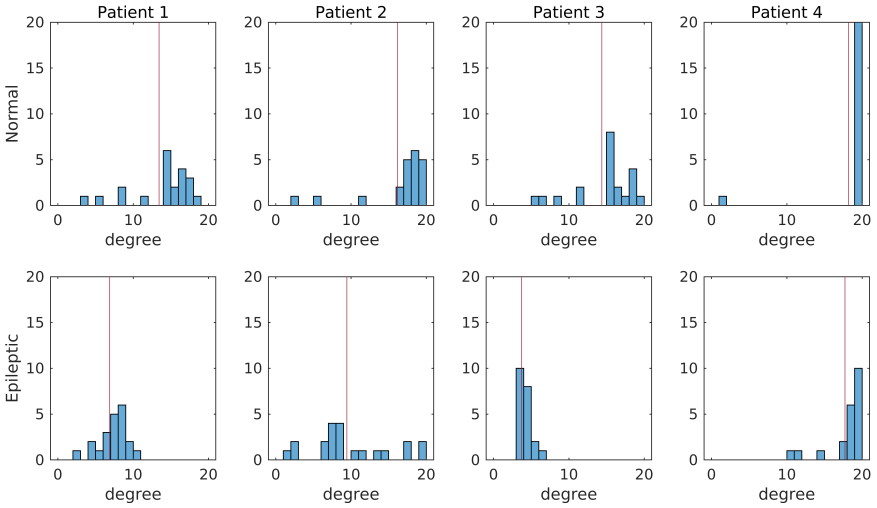


Figure 9: Degree distributions of the learned brain graphs from normal and epileptic EEG of different patients. The red vertical lines indicate the average node degree of the distributions.