TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Buciulea, A., Das, B., Isufi, E., & Marques, A. G. (2026). Learning Product Graphs from Two-Dimensional Stationary Signals. In *Proceedings of the ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 66-70). IEEE. https://doi.org/10.1109/ICASSP55912.2026.11465135

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11465135

LEARNING PRODUCT GRAPHS FROM TWO-DIMENSIONAL STATIONARY SIGNALS

Andrei Buciulea[⋆] *Bishwadeep Das*[†] *Elvin Isufi*[†] *Antonio G. Marques*[⋆]

[⋆] Dept. of Signal Theory and Communications, Rey Juan Carlos University, Madrid, Spain
[†] Delft University of Technology, Delft, The Netherlands

ABSTRACT

Graph learning aims to infer a network structure directly from observed data, enabling the analysis of complex dependencies in irregular domains. Traditional methods focus on scalar signals at each node, ignoring dependencies along additional dimensions such as time, configurations of the observation device, or populations. In this work, we propose a graph signal processing framework for learning graphs from *two-dimensional signals*, modeled as *matrix graph signals* generated by joint filtering along both dimensions. This formulation leverages the concept of graph stationarity across the two dimensions and leverages product graph representations to capture structured dependencies. Based on this model, we design an optimization problem that can be solved efficiently and provably recovers the optimal underlying Kronecker/Cartesian/strong product graphs. Experiments on synthetic data demonstrate that our approach achieves higher estimation accuracy and reduced computational cost compared to existing methods.

***Index Terms*—** Graph learning, network-topology inference, product graph, two-dimensional signals, graph-stationary signals

I. INTRODUCTION

Graphs are powerful for representing and analyzing signals on irregular domains. They model relationships between entities (nodes) and help extract insights from data (signals) observed over them. When the structure is unknown, graph learning (or network topology inference) aims to infer the graph directly from data. Early approaches were based on statistical methods [1, Ch. 7.3.1], including correlation- and partial-correlation-based techniques, which ultimately led to the development of the widely used graphical Lasso [1, 2]. Later, advances inspired by *graph signal processing (GSP)* introduced models that explicitly incorporated the relationship between graph structure and observed signals. Examples include smoothness-based models [3], sparse structural equation models [4], and stationarity-based approaches [5–7], which generalized classical models from regular domains (e.g., time series) to graphs. Building on these foundations, research has advanced toward more complex problems, considering the estimation of multiple graphs (time-varying graphs, task-specific, or online graph estimation) [8–12], handling perturbations in the graph and the signals (hidden nodes, missing data, or signal denoising) [13–16], and capturing higher-order interactions [17–19].

Despite this progress, most graph learning frameworks produce graphs that capture dependencies along only a single dimension between the entities considered as nodes. Here, each node has a scalar observation, and multiple realizations of such signals are available. These realizations may correspond to time samples, repetitions of an experiment, or data collected from multiple individuals (e.g., brain activity from different patients) [20–22]. Depending on the type of signals and the scenario considered, one or multiple graphs can be estimated from these realizations. However, in many applications, a more appropriate representation is a *two-dimensional signal*, where information is indexed both by the node (entities) and by the realization dimension (temporal, experimental, or population-level). Consequently, treating all realizations as a single dimension disregards the relationships across realizations, thereby limiting the interpretability and comprehensiveness of the learned graphs in contexts where dependencies across multiple dimensions are significant.

To model the dependencies across multiple dimensions simultaneously, recent research considers each observation to be indexed not only by a node (e.g., a movie) but also by an additional dimension such as time instants, experiments or users, resulting in two-dimensional or multiway graph signals. Thus, a (movie-user) node in a recommender system establishes the relationship between a user and a movie jointly as an entity. Several approaches have been developed to capture these structured dependencies. The *bigraphical Lasso* proposes an estimator for precision matrices based on the Cartesian product of graphs to jointly capture correlations across rows and columns [23], while *TERRALasso* generalizes this idea to multi-way tensor data with a sparse Kronecker-sum structure [24]. *EiGLasso* further improves scalability by exploiting eigendecomposition and Newton-based approximations for estimating sparse Kronecker-sum precision matrices [25], and *DNNLasso* ensures interpretability and computational efficiency via a diagonally nonnegative graphical Lasso [26]. Other works leverage smoothness priors and structured graph products—including Cartesian, Kronecker, and strong products—to learn graphs along multiple dimensions for capturing non-separable dependencies [27–32].

Our contribution. In this work, we adopt a GSP perspective to learn graphs from two-dimensional signals, assuming signal stationary with respect to the underlying network. Our contribution is to model stationary signals as the output of a joint filter that encodes the underlying graph structures along each dimension while acting on white Gaussian noise, thus, extending stationarity to two-dimensional graph signals and exploiting the Kronecker structure of the underlying graph. Based on this model, we develop an algorithm that leverages the Kronecker structure to learn graphs via a separable optimization problem which links to existing formulations for graph recovery from multidimensional signals. Finally, we validate our approach through extensive synthetic experiments against state-of-the-art baselines, demonstrating improvements in both estimation accuracy and computational efficiency.

Part of this work was supported by the TU Delft AI Labs programme, the NWO OTP GraSPA project (No. 19497), the NWO VENI project (No. 222.032), the Spanish AEI (10.13039/501100011033) under grant PID2022-136887NBI00, and the Community of Madrid within the ELLIS Unit Madrid framework, the IDEA-CM (TEC-2024/COM-89), and the CAM-URJC F1180 (CP2301) grants. Corresponding author: antonio.garcia.marques@urjc.es.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 10:38:03 UTC from IEEE Xplore. Restrictions apply.

II. PRELIMINARIES: LEARNING GRAPHS FROM ONE-DIMENSIONAL STATIONARY SIGNALS

A *graph signal* is a magnitude (feature) that varies over the nodes of a graph. In the graph-learning context, the goal is to infer the underlying graph structure (i.e., the pairwise relations among the nodes) from observed graph signals. Mathematically, this entails estimating $\mathbf{S} \in \mathbb{R}^{N\times N}$, the so-called *graph shift operator* (GSO) matrix, from R signals in $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_R] \in \mathbb{R}^{N\times R}$, with N denoting the number of nodes in the graph. The GSO encodes the graph topology and is commonly chosen as the adjacency matrix, the graph Laplacian, or their generalizations [33].

Stationary graph signals provide a model for linking data to graph structure. A stationary graph signal can be expressed as $\mathbf{x} = \mathbf{H}\mathbf{w}$, where $\mathbf{w}$ is standard normally distributed vector, and $\mathbf{H}$ is a *graph filter* of the form $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$, i.e., a polynomial in the GSO with coefficients $\{h_l\}$. The covariance matrix of a stationary signal, $\mathbf{C}_\mathbf{x} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top]$, is also a polynomial in $\mathbf{S}$ [34]. Consequently, $\mathbf{C}_\mathbf{x}$ and $\mathbf{S}$ share the same eigenvectors and commute [6, 15, 35], i.e., $\mathbf{C}_\mathbf{x}\mathbf{S} = \mathbf{S}\mathbf{C}_\mathbf{x}$. This motivates graph learning problem based on stationarity, where given the true covariance $\mathbf{C}_\mathbf{x}$, one seeks to recover a sparse GSO by solving

$$\min_{\mathbf{S}\in\mathcal{S}} \quad \|\mathbf{S}\|_1 \quad \text{s. t.} \quad \mathbf{C}_\mathbf{x}\mathbf{S} = \mathbf{S}\mathbf{C}_\mathbf{x}, \tag{1}$$

where the ℓ_1-norm promotes sparsity in the estimated graph, which is a typical assumption for real-world networks, and the commutativity constraint ensures that the signal is stationary on the estimated graph. $\mathcal{S}$ denotes the set of valid GSOs, e.g., adjacency matrices: $\mathcal{S} = \left\{\mathbf{S} \in \mathbb{R}^{N\times N} \,\middle|\, \mathbf{S} = \mathbf{S}^\top, S_{ij} \geq 0, S_{ii} = 0, \mathbf{S}\mathbf{1} = \mathbf{1}\right\}$, where the matrices are symmetric (undirected graphs), nonnegative, and have zero diagonal (no self-loops), and the last constraint avoids the trivial zero solution.

III. LEARNING GRAPHS FROM TWO-DIMENSIONAL STATIONARY SIGNALS

Two-dimensional graph signals naturally arise in applications such as spatio-temporal processes, images, or multi-modal data [36, 37]. Such signals can be represented as a matrix $\mathbf{Y} \in \mathbb{R}^{P\times Q}$, where the two dimensions capture distinct aspects of the data. For example, in spatio-temporal signals, P may correspond to spatial nodes and Q to time instances; more generally, the dimensions can represent any pair of modalities. When multiple realizations are available, the data can be organized as a tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{P\times Q\times R}$, with $\mathbf{Y}_r \in \mathbb{R}^{P\times Q}$ denoting the r-th slab of the tensor.

General vectorized model. If each (p, q) pair is considered as a node, the most general approach is to vectorize $\mathbf{Y}$ into $\mathbf{y} = \mathrm{vec}(\mathbf{Y}) \in \mathbb{R}^{PQ}$, consider a one-dimensional stationary graph signal model, and estimate the GSO $\mathbf{S}_{PQ} \in \mathbb{R}^{PQ\times PQ}$ without imposing any additional structure on the graph. Specifically, modelling the observations as stationary implies that $\mathbf{y} = \mathbf{H}_{PQ}\mathbf{w}$ with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{PQ})$, where $\mathbf{H}_{PQ}$ is a polynomial in the shift operator $\mathbf{S}_{PQ} \in \mathbb{R}^{PQ\times PQ}$ and $\mathbf{I}_{PQ} \in \mathbb{R}^{PQ\times PQ}$ denotes the identity matrix. In this formulation, $\mathbf{S}_{PQ}$ captures all possible relationships across both dimensions, treating each pair (p, q) as a node in the graph. The covariance $\mathbf{C}_\mathbf{y} = \mathbb{E}[\mathbf{y}\mathbf{y}^\top]$ commutes with $\mathbf{S}_{PQ}$, allowing the graph topology to be inferred by solving Problem (1). However, this approach suffers from poor scalability. First, the number of unknowns grows quadratically with PQ, which can be prohibitive even for estimation algorithms that run in polynomial time. Second, accurately estimating $\mathbf{C}_{PQ}$ is difficult, since the sample covariance $\frac{1}{R}\sum_{r=1}^{R} \mathrm{vec}(\mathbf{Y}_r)\mathrm{vec}(\mathbf{Y}_r)^T$ is typically low rank, requiring either a very large R or strong regularization.

Structured product-graph model. To address these limitations, we consider a structured approach by assuming that the observed signals are stationary on a product graph. Specifically, let $G_P = (\mathcal{V}_P, S_P)$ and $G_Q = (\mathcal{V}_Q, S_Q)$ be two graphs with $|\mathcal{V}_P| = P$ and $|\mathcal{V}_Q| = Q$ nodes, respectively, that capture relationships across dimensions P and Q. The product graph of G_P and G_Q, denoted by $\diamondsuit$, is defined as $G_{PQ} = G_P \diamondsuit G_Q = (\mathcal{V}, A_\diamondsuit)$, where $\diamondsuit$ represents either the Kronecker, Cartesian, or strong product [38]. A key property of all these constructions is that the eigenvectors of the product GSO are given by $\mathbf{V}_Q \otimes \mathbf{V}_P$, the Kronecker product of the eigenvectors of $\mathbf{S}_P$ and $\mathbf{S}_Q$ [38]. In our setting, this is relevant because the assumption of stationarity on the product graph implies that the covariance matrix $\mathbf{C}_\mathbf{y}$ shares its eigenvectors with the Kronecker product GSO $\mathbf{S}_{PQ} = \mathbf{S}_P \otimes \mathbf{S}_Q$. This observation motivates the following topology identification problem:

$$\begin{aligned} \min_{\mathbf{S}_P\in\mathcal{S},\, \mathbf{S}_Q\in\mathcal{S}} \quad & \|\mathbf{S}_P\|_1 + \beta\|\mathbf{S}_Q\|_1 \\ \text{s. t.} \quad & \mathbf{C}_\mathbf{y}(\mathbf{S}_Q \otimes \mathbf{S}_P) = (\mathbf{S}_Q \otimes \mathbf{S}_P)\mathbf{C}_\mathbf{y}, \end{aligned} \tag{2}$$

where the objective promotes sparse solutions by minimizing the sum of the ℓ_1 norms of $\mathbf{S}_P$ and $\mathbf{S}_Q$ with $\beta > 0$. Sparsity in both $\mathbf{S}_P$ and $\mathbf{S}_Q$ directly implies that the GSO of the product graph will also be sparse. The commutativity constraint between $\mathbf{C}_\mathbf{y}$ and $\mathbf{S}_P \otimes \mathbf{S}_Q$ follows from the assumption of stationarity on the product graph, since both matrices are simultaneously diagonalizable. Incorporating this additional structure reduces the number of optimization variables from P^2Q^2 (in the fully general case) to $P^2 + Q^2$, thereby improving tractability. Nonetheless, the problem remains nonconvex due to the Kronecker product in the commutativity constraint. Alternating optimization schemes are typically employed to address this and often yield good empirical results. However, recovery guarantees are difficult to establish, the large covariance matrix must still be estimated, and computations continue to involve handling large $PQ \times PQ$ matrices.

Dimension-wise covariance formulation. In this section, we adopt a more structured signal model that will lead to a more tractable optimization, bypassing some of the issues present when solving (2). The critical point is to assume that the two-dimensional signal is generated by a particular graph filtering process, as described next.

Assumption 1 *Given GSOs $\mathbf{S}_P \in \mathbb{R}^{P\times P}$ and $\mathbf{S}_Q \in \mathbb{R}^{Q\times Q}$, and polynomial graph filters $\mathbf{H}_P(\mathbf{S}_P)$ and $\mathbf{H}_Q(\mathbf{S}_Q)$, the two-dimensional graph signal $\mathbf{Y} \in \mathbb{R}^{P\times Q}$ is generated as*

$$\mathbf{Y} = \mathbf{H}_P(\mathbf{S}_P)\, \mathbf{W}\, \mathbf{H}_Q(\mathbf{S}_Q), \tag{3}$$

where the entries of the noise matrix $\mathbf{W} \in \mathbb{R}^{P\times Q}$ are i.i.d.

Here, $\mathbf{H}_P(\mathbf{S}_P)$ and $\mathbf{H}_Q(\mathbf{S}_Q)$ are polynomial filters in their respective GSOs, and for simplicity we refer to them as $\mathbf{H}_P$ and $\mathbf{H}_Q$. Assumption 1 states that the observations are generated through distinct filtering processes along each dimension. In a spatiotemporal setting, for instance, the dependencies across time may differ significantly from those across space, requiring different diffusions. Similarly, in recommender systems [39], the user and item dimensions may require different filters due to the heterogeneous nature of their underlying graphs. Moreover, the data generation model in (3) parallels the stationary time-vertex signal processing framework in [40], even though one of the two dimensions need not be temporal. If we interpret the Laplacian of the temporal graph in [40] as the shift operator in one dimension, the vectorized signals $\mathrm{vec}(\mathbf{Y})$ satisfy the criterion of joint wide-sense stationarity [40, Def. 1]. The matrix $\mathbf{Y}$

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 10:38:03 UTC from IEEE Xplore. Restrictions apply.

also satisfies the criterion of a jointly wide-sense stationary process, provided that both graphs are connected.[1]

Signals satisfying Assumption 1 clearly give rise to covariances diagonalized by $\mathbf{V}_Q \otimes \mathbf{V}_P$. To see this, note that vectorizing $\mathbf{Y}$ in (3) yields

$$\mathbf{y} = \big(\mathbf{H}_Q \otimes \mathbf{H}_P\big)\text{vec}(\mathbf{W}). \tag{4}$$

As a result, we have

$$\mathbf{C}_\mathbf{y} = \mathbb{E}[\mathbf{y}\mathbf{y}^\top] = \mathbf{H}_Q\mathbf{H}_Q^\top \otimes \mathbf{H}_P\mathbf{H}_P^\top = \mathbf{H}_Q^2 \otimes \mathbf{H}_P^2. \tag{5}$$

Since $\mathbf{H}_Q^2$ is a polynomial in $\mathbf{S}_Q$, its eigenvectors are $\mathbf{V}_Q$. Similarly, the eigenvectors of $\mathbf{H}_P^2$ are $\mathbf{V}_P$. Hence, the eigenvectors of $\mathbf{C}_\mathbf{y}$ in (5) are $\mathbf{V}_Q \otimes \mathbf{V}_P$. This implies that when signals follow the model in Assumption 1, the GSOs $\mathbf{S}_P$ and $\mathbf{S}_Q$ can be identified by solving (2).

Interestingly, the structure of (3) enables the design of a more efficient approach for estimating the GSO. To this end, we first present the following result as a proposition.

Proposition 1 *Let $\mathbf{Y}$ be a two-dimensional graph signal satisfying Assumption 1 and define matrices $\mathbf{C}_P = \mathbb{E}[\mathbf{Y}\mathbf{Y}^\top]$ and $\mathbf{C}_Q = \mathbb{E}[\mathbf{Y}^\top\mathbf{Y}]$. Then, it holds that:*

i) $\mathbf{C}_P = \text{tr}\big(\mathbf{H}_Q^2\big)\mathbf{H}_P^2$, which is a polynomial in $\mathbf{S}_P$.

ii) $\mathbf{C}_Q = \text{tr}\big(\mathbf{H}_P^2\big)\mathbf{H}_Q^2$, which is a polynomial in $\mathbf{S}_Q$.

Proof. See Appendix. ■

The result in the proposition allows us to compute (estimate) the covariance matrices separately and formulate the graph-learning problem as

$$\begin{aligned} \min_{\mathbf{S}_P\in\mathcal{S},\,\mathbf{S}_Q\in\mathcal{S}} \quad & \|\mathbf{S}_P\|_1 + \beta\|\mathbf{S}_Q\|_1 \\ \text{s.t.} \quad & \mathbf{C}_P\mathbf{S}_P = \mathbf{S}_P\mathbf{C}_P, \quad \mathbf{C}_Q\mathbf{S}_Q = \mathbf{S}_Q\mathbf{C}_Q. \end{aligned} \tag{6}$$

where the fact that $\mathbf{C}_P$ and $\mathbf{C}_Q$ are polynomials in $\mathbf{S}_P$ and $\mathbf{S}_Q$ is encoded in the (separate) commutativity constraints. This formulation is convex in $\{\mathbf{S}_P, \mathbf{S}_Q\}$ and separable, allowing each graph to be optimized independently. In addition, the number of constraints is significantly smaller than in (2), which reduces computational complexity. In practice, the covariances are estimated from R observations as $\hat{\mathbf{C}}_P = \sum_{r=1}^R \mathbf{Y}_r\mathbf{Y}_r^\top$ and $\hat{\mathbf{C}}_Q = \sum_{r=1}^R \mathbf{Y}_r^\top\mathbf{Y}_r$. Since these covariance matrices are now much smaller, modest values of R are sufficient to obtain reliable estimates. The formulation in (6) is a special case of eq (45) in [5] for $n = 2$, $\alpha_1 = 0.5$, and $\alpha_2 = 2\beta$.

Having established the different problem formulations, we now summarize them along with their corresponding solution strategies. Problem (1) is convex in $\mathbf{S}$ and can be solved using standard off-the-shelf solvers or more efficient methods such as ADMM [41], resulting in an unstructured learned graph. The main drawback in this case is the high dimensionality, which complicates both the computational complexity and the accurate estimation of covariance matrices. In contrast, Problem (2) imposes a product graph structure. This reduces the number of optimization unknowns, but introduces a non-convex (Kronecker product) constraint that must be handled via alternating optimization [42]. While this approach lowers computational complexity, the estimation of the high-dimensional covariance matrix remains an issue. Finally, by leveraging additional structure in the graph signals, Problem (6) yields a convex, low-dimensional optimization that can be solved separately for each dimension, thereby bypassing the need to estimate the full covariance matrix.

[1]The proofs follow directly from [40].

IV. NUMERICAL EXPERIMENTS

We evaluate the performance of the proposed formulations on synthetic data and compare them against existing approaches from the literature. The methods considered are: spectral templates (ST), which solves the problem in (1); Kronecker-ST (K-ST), which addresses the problem in (2); separate Kronecker-ST (SepK-ST), which solves (6); DNNLasso, the method proposed in [26]; and TERRALasso, the method introduced in [24]. For evaluation, we focus on unweighted graphs and employ the F-score, defined as

$$\text{F-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{7}$$

where $precision$ denotes the percentage of estimated edges that are edges of the ground-truth graph and $recall$ the percentage of existing edges that were correctly estimated. In all experiments, both $\mathbf{S}_P$ and $\mathbf{S}_Q$ are generated as Erdős-Rényi random graphs with a node connection probability of 0.3. Simulations are repeated over 100 independent graph realizations, and the mean F-score is reported. Implementations of the proposed approaches are carried out in `CVX`. More specific details about the implementation setup of the experiments can be found in the available code on GitHub[2].

Accuracy vs. time complexity. We compare ST, K-ST, and SepK-ST in terms of graph recovery accuracy and computational runtime. Fig. 1a) reports the F-score and runtime as the graph size increases, under the assumption of perfect covariance knowledge. From the F-score results (solid lines), we observe that both K-ST and SepK-ST achieve identical recovery performance and clearly outperform ST, which fails to correctly capture the underlying structure and exhibits significantly lower F-scores. Both K-ST and SepK-ST explicitly exploit the two-dimensional structure of the signals, enabling accurate recovery of the Kronecker-structured graphs, whereas ST ignores this structure. Regarding runtime (dotted lines), all methods become slower as the graph increases, which is consistent with the higher complexity of larger problems. However, ST scales much worse than K-ST and SepK-ST as it requires estimating $(PQ)^2$ variables, whereas K-ST and SepK-ST only involve $P^2 + Q^2$ variables. Between K-ST and SepK-ST, the difference lies in implementation: although they optimize over the same number of variables, K-ST requires large matrix operations because of Kronecker products, which increases runtime and makes it slower than SepK-ST. This highlights the importance of leveraging the inherent structure of two-dimensional signals. SepK-ST provides a good balance between recovery performance and runtime.

Accuracy vs. commutativity. We now evaluate the behavior of ST, K-ST, and SepK-ST when the covariance matrix is estimated from a limited number R of signals. Fig. 1b) reports the F-score and commutativity of the estimated graphs as R increases, fixing the graph size to $N = 64$ nodes with $P = Q = 8$. The F-scores (solid lines) show that SepK-ST outperforms ST and K-ST, achieving better graph recovery across different R. This is partly because smaller graphs require fewer samples for accurate estimation, while ST must handle a much larger graph and thus needs more samples. K-ST relies on the covariance of the vectorized two-dimensional signals, which also requires more samples to reach performance comparable to SepK-ST. We also assess commutativity, defined as $\|\mathbf{C}\hat{\mathbf{S}} - \hat{\mathbf{S}}\mathbf{C}\|_F^2$, where $\hat{\mathbf{S}}$ is the estimated Kronecker graph, i.e., how well the estimated graphs satisfy stationarity (lower implies better). ST consistently yields lower values than K-ST and SepK-ST, indicating that it identifies graphs on which the signals are stationary, but these

[2]https://github.com/andreibuciulea/twodimST

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 10:38:03 UTC from IEEE Xplore. Restrictions apply.

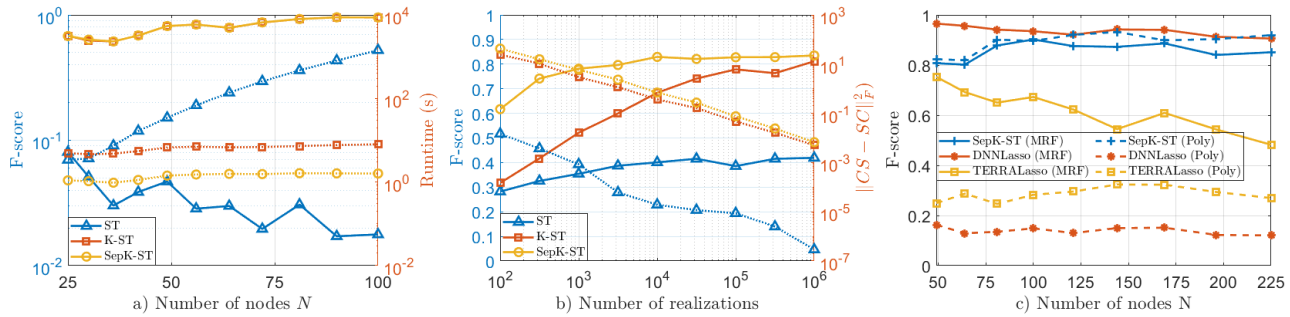


Fig. 1: Performance comparison of three proposed approach variants. a) F-score (solid lines, left y-axis) and runtime in seconds (dotted lines, right y-axis) plotted against the total number of nodes. b) F-score (solid lines, left y-axis) and commutativity values of the estimated graphs (dotted lines, right y-axis) as a function of the number of samples. c) F-score comparison of the proposed approach (SepK-ST) against two methods from the literature, as the graph size increases, under two different signal generation models: $\mathbf{C}_{MRF}$ and $\mathbf{C}_{Poly}$.

graphs fail to capture the true two-dimensional structure. In contrast, SepK-ST shows decreasing commutativity values alongside increasing F-score as R grows, demonstrating its ability to enforce stationarity while recovering model-consistent graphs. Lastly, K-ST attains lower commutativity values than SepK-ST, but the recovered graphs deviate more from the ground truth, suggesting that K-ST requires more samples to match SepK-ST. This discrepancy shrinks as R increases, reducing the F-score performance gap. Overall, the results indicate that SepK-ST is the most reliable approach in sample-limited regimes, balancing stationarity and accurate recovery.

Comparing baselines. Here, we compare the proposed approaches against two state-of-the-art algorithms from the literature that also account for two-dimensional graph signals with product graph structure. To ensure fairness, we generate signals using two distinct models whose true covariances are given by $\mathbf{C}_{MRF} = (\alpha\mathbf{I} + \mathbf{S})^{-1}$ and $\mathbf{C}_{Poly} = \left(\sum_{l=1}^{L} h_l \mathbf{S}^l\right)^2$, with $\alpha > 0$ sufficiently large to guarantee positive definiteness and $L = 3$. The number of samples is fixed to 10^6. The F-score results (solid lines in Fig. 1c) show that DNNLasso achieves the best performance, closely followed by SepK-ST. TERRALasso is also able to recover the graph, but its accuracy decreases significantly as the graph size grows. This happens because $\mathbf{C}_{MRF}$ aligns perfectly with the modeling assumptions of DNNLasso, explaining its superior recovery performance. When signals are generated obeying $\mathbf{C}_{Poly}$, our method is the only one capable of consistently recovering the underlying graph while maintaining high F-score as the number of nodes increases. In contrast, DNNLasso and TERRALasso fail to capture the structure under this scenario. This suggests that our approach is not only accurate and computationally efficient, but it is also verstaile as it generalizes to graphs arising from different two-dimensional data models.

V. CONCLUSIONS

In this work, we introduced a novel framework for learning graphs from two-dimensional stationary signals under a GSP perspective. After discussing a generic stationarity-based formulation, we developed a lower-complexity approach that models the overall graph as the Kronecker/Cartesian/strong product of two separate graphs. Finally, by imposing additional structure and modeling the observed signals as the output of a separate graph filtering across dimensions, we established explicit connections between dimension-specific covariances and their underlying graph shift operators. This allowed us to reformulate the topology identification task as two separable, tractable convex problems, each defined over a single dimension, while preserving the Kronecker structure of the model. Numerical experiments further demonstrated that the proposed approach achieves high recovery accuracy, scales more efficiently with graph size, and outperforms existing baselines in both accuracy and runtime.

VI. APPENDIX

Proof of Proposition (1)**.** We begin by vectorizing (1). This yields

$$\mathbf{y} = (\mathbf{H}_Q \otimes \mathbf{H}_P)\,\mathrm{vec}(\mathbf{W}),$$

so that the theoretical covariance of $\mathbf{y}$ is

$$\mathbf{C}_\mathbf{y} = \mathbb{E}[\mathbf{y}\mathbf{y}^\top] = \mathbf{H}_Q^2 \otimes \mathbf{H}_P^2. \tag{8}$$

Matrix-form analysis. Alternatively, let us work directly with the matrix $\mathbf{Y}$. The covariance along dimension P is

$$\mathbf{C}_P = \mathbb{E}[\mathbf{Y}\mathbf{Y}^\top] = \mathbb{E}[\mathbf{H}_P\mathbf{W}\mathbf{H}_Q\mathbf{H}_Q^\top\mathbf{W}^\top\mathbf{H}_P^\top].$$

Since

$$\mathbb{E}[\mathbf{W}\mathbf{H}_Q\mathbf{H}_Q^\top\mathbf{W}^\top] = \mathrm{trace}(\mathbf{H}_Q^2)\,\mathbf{I},$$

it follows that

$$\mathbf{C}_P = \mathrm{trace}(\mathbf{H}_Q^2)\,\mathbf{H}_P^2, \qquad \mathbf{C}_Q = \mathrm{trace}(\mathbf{H}_P^2)\,\mathbf{H}_Q^2. \tag{9}$$

Relation between covariances. Taking the Kronecker product of the two covariance matrices, we obtain

$$\begin{aligned}\mathbf{C}_Q \otimes \mathbf{C}_P &= \mathrm{trace}(\mathbf{H}_Q^2)\,\mathrm{trace}(\mathbf{H}_P^2)\,(\mathbf{H}_Q^2 \otimes \mathbf{H}_P^2)\\ &= \|\mathbf{H}_Q\|_F^2\,\|\mathbf{H}_P\|_F^2\,\mathbf{C}_\mathbf{y},\end{aligned} \tag{10}$$

which establishes the connection between $\mathbf{C}_Q \otimes \mathbf{C}_P$ and $\mathbf{C}_\mathbf{y}$.

Spectral characterization. Let the eigendecompositions of the shift operators be

$$\mathbf{S}_Q = \mathbf{V}_Q\mathbf{\Lambda}_Q\mathbf{V}_Q^\top, \qquad \mathbf{S}_P = \mathbf{V}_P\mathbf{\Lambda}_P\mathbf{V}_P^\top.$$

It follows that

$$\mathbf{H}_Q^2 = \mathbf{V}_Q\mathbf{H}(\mathbf{\Lambda}_Q)^2\mathbf{V}_Q^\top, \qquad \mathbf{H}_P^2 = \mathbf{V}_P\mathbf{H}(\mathbf{\Lambda}_P)^2\mathbf{V}_P^\top,$$

where $\mathbf{H}(\mathbf{\Lambda}_Q)$ and $\mathbf{H}(\mathbf{\Lambda}_P)$ are diagonal matrices containing the frequency responses.

Substituting these into (9), we see that $\mathbf{C}_Q$ and $\mathbf{C}_P$ share eigenvectors with $\mathbf{S}_Q$ and $\mathbf{S}_P$, respectively. Likewise, substituting into (10) gives

$$\mathbf{C}_Q \otimes \mathbf{C}_P = (\mathbf{V}_Q \otimes \mathbf{V}_P)\left(\mathbf{H}(\mathbf{\Lambda}_Q)^2 \otimes \mathbf{H}(\mathbf{\Lambda}_P)^2\right)(\mathbf{V}_Q^\top \otimes \mathbf{V}_P^\top). \tag{11}$$

Therefore, $\mathbf{C}_Q \otimes \mathbf{C}_P$ commutes with $\mathbf{S}_Q \otimes \mathbf{S}_P$. ■

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 10:38:03 UTC from IEEE Xplore. Restrictions apply.

VII. REFERENCES

[1] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. New York, NY: Springer, 2009.

[2] N. Meinshausen and P. Buhlmann, "High-dimensional graphs and variable selection with the lasso," *Ann. Statist.*, vol. 34, pp. 1436–1462, 2006.

[3] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.

[4] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLoS computational biology*, vol. 9, no. 5, p. e1003068, 2013.

[5] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, 2017.

[6] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*. IEEE, 2017, pp. 6523–6527.

[7] A. Buciulea, J. Ying, A. G. Marques, and D. P. Palomar, "Polynomial graphical lasso: Learning edges from Gaussian graph-stationary signals," *IEEE Trans. Signal Process.*, vol. 73, pp. 1153–1167, 2025.

[8] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, March 2017, pp. 2826–2830.

[9] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *J. of the Roy. Statistical Soc.: Ser. B (Statistical Methodology)*, vol. 76, no. 2, pp. 373–397, 2014.

[10] M. Navarro, Y. Wang, A. G. Marques, C. Uhler, and S. Segarra, "Joint inference of multiple graphs from matrix polynomials," *J. Mach. Learn. Res. (JMLR)*, vol. 23, no. 1, pp. 3302–3336, 2022.

[11] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "Online learning of time-varying signals and graphs," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*. IEEE, 2021, pp. 5230–5234.

[12] A. Buciulea, M. Navarro, S. Rey, S. Segarra, and A. G. Marques, "Online network inference from graph-stationary signals with hidden nodes," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*. IEEE, 2025, pp. 1–5.

[13] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, "Latent variable graphical model selection via convex optimization," *Ann. of Statist.*, vol. 40, no. 4, pp. 1935–1967, 2012.

[14] A. Chang, T. Yao, and G. I. Allen, "Graphical models and dynamic latent factors for modeling functional brain connectivity," in *IEEE Data Science Wrksp. (DSW)*, 2019, pp. 57–63.

[15] A. Buciulea, S. Rey, and A. G. Marques, "Learning graphs from smooth and graph-stationary signals with hidden variables," *IEEE Trans. Signal Process.*, vol. 8, pp. 273–287, 2022.

[16] Z. Zhang and Z. Zhao, "A joint graph signal and Laplacian denoising network," in *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2024, pp. 1–5.

[17] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Trans. Signal Process.*, vol. 68, pp. 2992–3007, 2020.

[18] J. G. Young, G. Petri, and T. P. Peixoto, "Hypergraph reconstruction from network data," *Communications Physics*, vol. 4, no. 1, p. 135, 2021.

[19] A. Buciulea, E. Isufi, G. Leus, and A. G. Marques, "Learning graphs and simplicial complexes from data," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*. IEEE, 2024, pp. 9861–9865.

[20] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, "Network inference via the time-varying graphical lasso," *J. Mach. Learn. Res. (JMLR)*, vol. 18, no. 1, pp. 3319–3370, 2017.

[21] V. Kalofolias, "How to learn a graph from smooth signals," in *Intl. Conf. Artif. Intel. Statist. (AISTATS)*, 2016, pp. 920–929.

[22] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning graphs from data: A signal representation perspective," in *IEEE Signal Process. Mag.*, vol. 36, no. 3. IEEE, 2019, pp. 44–63.

[23] A. Kalaitzis, J. Lafferty, N. D. Lawrence, and S. Zhou, "The bigraphical lasso," in *Intl. Conf. on Mach. Learn.* PMLR, 2013, pp. 1229–1237.

[24] K. Greenewald, S. Zhou, and A. Hero, "Tensor graphical lasso (TeraLasso)," *IEEE Trans. Signal Process.*, vol. 67, no. 14, pp. 3706–3721, 2019.

[25] K. Greenewald and A. Hero, "EiGLasso for sparse inverse covariance estimation with Kronecker product structure," *IEEE Trans. Signal Process.*, vol. 67, no. 2, pp. 514–529, 2019.

[26] M. Lin and Y. Zhang, "DNNLasso: Scalable graph learning for matrix-variate data," in *Intl. Conf. Artif. Intell. Stat. (AISTATS)*. PMLR, 2024, pp. 316–324.

[27] M. A. Lodhi and W. U. Bajwa, "Learning product graphs underlying smooth graph signals," *arXiv preprint arXiv:2002.11277*, 2020.

[28] C. Shi and G. Mishne, "Learning Kronecker-structured graphs from smooth signals," *arXiv preprint arXiv:2505.09822*, 2025.

[29] S. K. Kadambari and S. P. Chepuri, "Learning product graphs from multidomain signals," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*. IEEE, 2020, pp. 5665–5669.

[30] C. Shi and G. Mishne, "Learning Cartesian product graphs with Laplacian constraints," in *Intern. Conf. Art. Int. Stat.* PMLR, 2024, pp. 2521–2529.

[31] S. K. Kadambari and S. P. Chepuri, "Product graph learning from multi-domain data with sparsity and rank constraints," *IEEE Trans. Signal Process.*, vol. 69, pp. 5665–5680, 2021.

[32] C. Zhang, Y. He, and H.-T. Wai, "Product graph learning from multi-attribute graph signals with inter-layer coupling," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*. IEEE, 2023, pp. 1–5.

[33] P. Djuric and C. Richard, *Cooperative and Graph Signal Processing: Principles and Applications*. Academic Press, 2018.

[34] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, 2017.

[35] R. Shafipour and G. Mateos, "Online topology inference from streaming stationary graph signals with partial connectivity information," *Algorithms*, vol. 13, no. 9, p. 228, 2020.

[36] Y. Yan, J. Hou, Z. Song, and E. E. Kuruoglu, "Signal processing over time-varying graphs: A systematic review," *arXiv preprint arXiv:2412.00462*, 2024.

[37] J. Stanley, E. Chi, and G. Mishne, "Multiway graph signal processing on tensors: Integrative analysis of irregular geometries," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 160–173, 2020.

[38] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Sig. Proc. Mag.*, vol. 31, no. 5, pp. 80–90, 2014.

[39] W. Huang, A. G. Marques, and A. Ribeiro, "Rating prediction via graph signal processing," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5066–5081, 2018.

[40] A. Loukas and N. Perraudin, "Stationary time-vertex signal processing," *EURASIP journal on advances in signal processing*, vol. 2019, no. 1, pp. 1–19, 2019.

[41] S. Boyd, N. Parikh, E. Chu, B.Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, 2011.

[42] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Mathematical methods of operations research*, vol. 66, pp. 373–407, 2007.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 12,2026 at 10:38:03 UTC from IEEE Xplore. Restrictions apply.