

# Joint Network Topology Inference Via a Shared Graphon Model

Madeline Navarro<sup>✉</sup>, *Graduate Student Member, IEEE*, and Santiago Segarra<sup>✉</sup>, *Senior Member, IEEE*

**Abstract**—We consider the problem of estimating the topology of multiple networks from nodal observations, where these networks are assumed to be drawn from the same (unknown) random graph model. We adopt a graphon as our random graph model, which is a nonparametric model from which graphs of potentially different sizes can be drawn. The versatility of graphons allows us to tackle the joint inference problem even for the cases where the graphs to be recovered contain different number of nodes and lack precise alignment across the graphs. Our solution is based on combining a maximum likelihood penalty with graphon estimation schemes and can be used to augment existing network inference methods. The proposed joint network and graphon estimation is further enhanced with the introduction of a robust method for noisy graph sampling information. We validate our proposed approach by comparing its performance against competing methods in synthetic and real-world datasets.

**Index Terms**—Network topology inference, graph learning, joint inference, graphon.

## I. INTRODUCTION

NETWORKS conveniently capture systems with complicated relationships and intuitively represent structure via dyadic connections. Data consisting of entities in interconnected systems, tangible or abstract, are ubiquitous in multiple fields. Network structures are highly utilized across these many disciplines for representation and analysis of complex information [2], such as ecology for predicting animal behavior [3], neuroscience for modeling relationships between neurons [4], and environmental science for discovering and predicting outcomes of climate relationships [5]. Interpretation of networks varies greatly depending on the application. For example, networks may represent physical systems, such as road networks or joint connectivity for skeletal data [6]. In contrast, the nature of a connected system may be more abstract, as is the case for correlation networks, where connectivity represents statistical

Manuscript received 27 April 2022; revised 5 September 2022 and 4 November 2022; accepted 9 November 2022. Date of publication 21 November 2022; date of current version 6 December 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ketan Rajawat. This work was supported by NSF under Award CCF-2008555. An earlier version were presented at ICASSP 2022 [DOI: 10.1109/ICASSP43922.2022.9746332]. (Corresponding author: Madeline Navarro.)

The authors are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: nav@rice.edu; segarra@rice.edu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TSP.2022.3223523>, provided by the authors.

Digital Object Identifier 10.1109/TSP.2022.3223523

interdependencies between observed variables. Additionally, networks may or may not be directly observable. As an intuitive example, consider that structural (anatomical) connectivity between neurons can be directly measured [7], whereas functional connectivity between brain regions cannot be known but must be estimated by observing neural responses [4], [8].

While networks are convenient and interpretable tools for tasks on complex data, the underlying structure may be unavailable. For instance, the true structure of unobservable networks cannot be provided but must be obtained, as with brain functional networks [4], [8], the correlations in social behavior between animals [3], or any abstract network where we cannot measure connectivity patterns directly. Alternately, the underlying network may be expensive to obtain, as with brain structural connectivity [7]. The ubiquitous problem of recovering network connectivity from graph measurements has been well studied in fields such as statistics [9] and signal processing [10]. Given data in the form of nodal observations, network connectivity via data-driven methods include graphical models [11], [12], structural equation models [13], and graph signal processing (GSP)-based approaches [10], [14], [15], [16].

In the case of inferring the topologies of multiple networks, separate estimation is a feasible methodology. However, in many scenarios a joint inference method may achieve better performance by leveraging common structures between the graphs to be inferred. For instance, one would expect certain levels of similarities between the brain networks of different healthy individuals or between the same social network observed at different points in time. In this paper, we consider the prevalent problem of inferring the topology of multiple networks while assuming that networks share structural similarities. Many applications rely on multiple instances of interconnected relationships observed over time or in several scenarios, and these complex structures can be conveniently represented by a set of networks. E.g., Brain networks, functional or structural, are valuable tools for diagnosis, and estimating multiple networks is necessary for analysis of many patients or scenarios [4]. One of the most prominent scenarios requiring the acquisition of multiple networks is when networks vary over time [8]. For example, we would expect that the social network of a species of interest will evolve over time [17].

We consider the problem of *recovering the connectivity of multiple networks* whose structures are represented by graphs assumed to be *sampled from the same (unknown) random graph model*. We adopt the nonparametric network model as

a graphon [18], but the specific graphon model need not be known a priori for our proposed method. While estimation of multiple networks is well-studied, to the best of our knowledge we provide the only work in the literature that utilizes a shared graphon relationship to jointly estimate graphs of potentially different sizes.

The contributions of our paper are as follows:

- 1) We present a methodology to *infer multiple networks that potentially lack node alignment and may have different sizes* by leveraging the assumption that graphs come from the same nonparametric network model.
- 2) We detail how our approach can be combined with existing network inference methods, effectively providing a whole family of methods to solve the problem of interest.
- 3) We develop a robust version of the problem where the exact sampling criteria of the graphs from the graphon are not known, but we are instead given noisy sampling information.
- 4) Through numerical experiments in synthetic and real-world data we demonstrate the performance of our method in comparison with separate inference and competing joint inference algorithms.

The remainder of this paper is organized as follows. We review graph signal processing and graphons in Section II, along with past related work. We introduce the problem of interest in Section III. Our proposed problem formulations are presented in Section IV, and the algorithm development is discussed in Section V. We expand the problem to a noisy graph sampling setting and present a robust solution in Section VI. Section VII presents experimental results of all proposed algorithms on synthetic and real-world data. Finally, we close with conclusions and discussions of future directions in Section VIII.

## II. PRELIMINARIES

### A. Notation

The following notation will be used in this paper. We represent vectors as bold lowercase letters  $\mathbf{x}$ , entries of which are indexed by  $x_i$ . Matrices are bold uppercase letters  $\mathbf{X}$ , where entries are indexed by  $X_{ij}$ , and  $\mathbf{X}_i$  represents the  $i$ -th row of  $\mathbf{X}$ . The superscript  $\top$  denotes the transpose. A matrix with a calligraphic letter subscript  $\mathbf{X}_{\mathcal{I}}$  denotes the submatrix of  $\mathbf{X}$  consisting of rows of  $\mathbf{X}$  indexed by the set  $\mathcal{I}$ . The notation  $\mathbf{X}_{\mathcal{I}}^{\top}$  is ordered by first selecting the rows indexed by  $\mathcal{I}$  then transposing the result. Thus, selecting the  $i$ -th column of a matrix  $\mathbf{X}$  is represented by  $[\mathbf{X}^{\top}]_i^{\top}$  and columns of  $\mathbf{X}$  indexed by  $\mathcal{I}$  is denoted by  $[\mathbf{X}^{\top}]_{\mathcal{I}}^{\top}$ . We define three special index sets  $\mathcal{L}$ ,  $\mathcal{U}$ , and  $\mathcal{D}$  referring to the lower triangle, upper triangle, and diagonal indices of a square matrix. A square matrix  $\mathbf{X} \in \mathbb{R}^{N \times N}$  with the subscript  $\mathcal{L}$  as  $\mathbf{X}_{\mathcal{L}}$  returns a column vector of length  $N(N - 1)/2$  of the vertical concatenation of the lower triangular entries of  $\mathbf{X}$ . The sets  $\mathcal{U}$  and  $\mathcal{D}$  return similar column vectors. We let  $\mathbf{I}_N$  and  $\mathbf{1}_N$  represent the identity matrix of size  $N \times N$  and the all-ones column vector of length  $N$ . The Kronecker product, the Kronecker sum, and the Hadamard product are denoted by  $\otimes$ ,  $\oplus$ , and  $\circ$ , respectively.

We use  $\text{vec}(\mathbf{X})$  to represent the column vector containing the vertical concatenation of the columns in  $\mathbf{X}$ .

### B. Graph Signal Processing

We consider undirected, unweighted graphs of the form  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with node (vertex) set  $\mathcal{V}$  of cardinality  $N$  and edge set  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ . The structure of a graph can be represented by its adjacency matrix  $\mathbf{S} \in \{0, 1\}^{N \times N}$ , where  $S_{ij} \neq 0$  if and only if the edge  $(i, j)$  exists in the network, and  $S_{ij} = 0$  otherwise. We define graph signals as real-valued observations at each of the  $N$  nodes, represented by a vector  $\mathbf{x} \in \mathbb{R}^N$ . We may associate these nodal values with the graph topology via graph signal models. Choices for graph signal models include stationary signals that result from diffusion processes over the graph [19], [20], [21] or as multivariate random numbers, where the graph structure represents statistical dependencies between variables [11], [12].

### C. Graphons

A graphon is a bounded symmetric measurable function  $\mathcal{W} : [0, 1]^2 \rightarrow [0, 1]$  whose domain can be interpreted as edges in an infinitely large adjacency matrix, while the range of  $\mathcal{W}$  represents edge probabilities. By this definition, a graphon can be seen as a random graph model from which graphs with similar structural characteristics can be sampled [18], [22], [23]. Generating an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  from a graphon  $\mathcal{W}$  consists of two steps: (1) selecting a random value between 0 and 1 for each node, and (2) assigning an edge between nodes with probability equal to the value of the graphon at the their randomly sampled points. Formally, the steps are as follows

$$\zeta_i \sim \text{Uniform}([0, 1]) \quad \forall i \in \mathcal{V}, \quad (1a)$$

$$S_{ij} = S_{ji} \sim \text{Bernoulli}(\mathcal{W}(\zeta_i, \zeta_j)) \quad \forall (i, j) \in \mathcal{V} \times \mathcal{V}, \quad (1b)$$

where the latent variables  $\zeta_i \in [0, 1]$  are independently drawn for each node  $i$ . This notion of graphon encompasses many commonly used exchangeable distributions on networks. Indeed, Erdős-Rényi graph models are represented via constant graphons [24] and stochastic block models (SBMs) via piecewise-constant graphons [25].

In our case, we assume that graphs are sampled from the same graphon, which is also unknown. Therefore, we propose a method to jointly estimate both the graphs and the underlying graphon.

### D. Related Work

**Joint network inference.** Inferring multiple networks has been well-studied, particularly for graphical model estimation, but most methods require strong assumptions about sizes and node alignment across networks. Statistical methods that estimate multiple graphical models typically involve modifications of joint graphical lasso with novel penalties encouraging similarity among networks [26], [27], [28], [29], [30]. While far fewer than their statistical counterpart, GSP-based methods for inferring multiple networks are prevalent [31], [32], [33], [34]. A particularly prominent scenario of joint network inference is

estimation of time-varying graphs, where typical assumptions include slow or smooth network changes over time [35], [36], [37], [38] or smooth graph signals, that is, signal values across nodes vary little [39], [40], [41], [42]. A more challenging problem is the inference of multiple networks where it is not known from which network observed data are generated. We note GSP methods [32], [43], [44], [45] and statistical methods [46], [47], [48] within this category. We refer the reader to the reviews in [10], [49], [50] for more examples of multiple network estimation.

The majority of joint network inference methods require network estimation on the same node set (and, thus, of the same size). One exception includes brain network estimation with brain regions of different coarseness [51], resulting in potentially different sized networks; however, we not only allow for regions at possibly different coarseness levels, but networks may consist of completely different sets of regions altogether.

Furthermore, assumptions on similarity between graph structures mostly concentrate on edgewise relationships, with exceptions including [52], [53], [54], [55] that encourage similar structural characteristics without relying on edgewise comparisons in particular. Similarly to our paper, [52] introduced a random graph model for estimating networks that share a common subspace without encouraging similar edge structures. Unlike our approach, all of these methods require networks to have the same size.

**Graphon estimation.** The proposed estimation procedure jointly estimates a set of graphs along with their unknown shared graphon. Graphon estimation is a well-studied problem [56], [57], where the random graph model is estimated from a binary adjacency matrix. Methods include estimating the graphon as a continuous smooth object [58], [59], [60], [61] along with the coarser SBM estimation [62], [63], [64], [65], [66], [67]. In many cases, the entire graphon is not needed but only the probability matrix from which the graphon was sampled, i.e., the value of the graphon  $\mathcal{W}(x, y)$  only at the latent sample points  $(x, y) \in \zeta \times \zeta$ , and these methods include neighborhood smoothing [68], [69], [70], matrix completion [71], [72], [73], or low-rank approximations [74]. In this work, we present a framework that can leverage any graphon estimation method for network estimation.

While most graphon estimation methods assume availability of only one sampled adjacency matrix, a few works exist that estimate graphon models from multiple graphs [62], [65], [68]. In our case, we infer a graphon from multiple networks without prior knowledge of the network structure.

### III. PROBLEM STATEMENT

Leveraging the proposed graphon relationship among networks, we present an algorithm that simultaneously estimates heterogeneous networks and accounts for a common graphon model. We consider undirected, unweighted graphs without self-loops and sampled from a shared graphon. Formally, consider a set of  $K$  different graphs  $\{\mathcal{G}^{(k)}\}_{k=1}^K$  where the  $k$ -th graph has  $N^{(k)}$  nodes. The set of undirected, unweighted adjacency matrices is represented by the set of adjacency matrices

$\{\mathbf{S}^{(k)}\}_{k=1}^K$ . Assume also that there is a set of graph signals provided for each graph, represented by  $\mathbf{X}^{(k)} := [\mathbf{x}_1^{(k)} \cdots \mathbf{x}_{r_k}^{(k)}] \in \mathbb{R}^{N^{(k)} \times r_k}$ , where the  $r_k$  columns contain the graph signals corresponding to the  $k$ -th graph. We further assume that all graphs are sampled from the same generative model, a graphon  $\mathcal{W}$ . With some abuse of notation, we let  $\mathbf{S}$  and  $\mathbf{X}$  represent the sets of adjacency matrices  $\mathbf{S} = \{\mathbf{S}^{(k)}\}_{k=1}^K$  and graph signals  $\mathbf{X} = \{\mathbf{X}^{(k)}\}_{k=1}^K$ , respectively. We present our problem as follows.

**Problem 1:** Given sets of observations  $\mathbf{X} = \{\mathbf{X}^{(k)}\}_{k=1}^K$  for  $K$  graphs, find the adjacency matrices  $\mathbf{S} = \{\mathbf{S}^{(k)}\}_{k=1}^K$  under the assumptions that (AS1) all graphs are sampled from the same (unknown) graphon  $\mathcal{W}$  and (AS2) the latent point sets  $\zeta^{(k)}$  in (1a) for each graph are known.

The first assumption (AS1) creates a relationship among the graphs, and with it we may improve estimation of graphs by jointly inferring the graph structures given their shared relationship. The second assumption (AS2) eliminates the identifiability problem for graphon estimation, where multiple graphons can lead to the same random graph distribution [22]. When all latent point sets are equivalent, i.e.,  $\zeta^{(k)} = \zeta$  for all graphs  $k \in \{1, 2, \dots, K\}$ , (AS2) is equivalent to the assumption in previous joint network inference methods, where node alignment is present and known for all pairs of graphs. However, assuming possibly different known latent point sets is a weaker assumption than that of previous methods, as we do not require node alignment for the graphs. Furthermore, in Section VI we relax the assumption (AS2) where the latent point sets are not exactly known and only noisy sets are available.

The assumption (AS2) corresponds intuitively to situations of known sensor placement, such as known locations of electrode placement for neural response data collection or known climate regions to be observed. For example, the brain functional networks of multiple subjects may be measured by considering the same known brain regions or neurons across subjects [4]. Inferred graphs may also correspond to statistical interdependence between pairs of variables in a climate data set, where variables are measured at known spatial regions of earth [5].

## IV. GRAPHON-AIDED JOINT NETWORK ESTIMATION

In Sections IV-A and IV-B we tackle two versions of Problem 1 of increasing difficulty whereas in Section IV-C we explain how these solutions can be combined with existing network inference methods.

### A. Graphs and Probability Matrix Estimation

First consider the case where all graphs are sampled as in (1a) from the same points in the graphon space, that is,  $\zeta^{(k)} = \zeta$  for all  $k \in \{1, 2, \dots, K\}$ ; see Fig. 1(a). In practice, this case arises, e.g., when using the same sensor placement under multiple trials or experiments. Since we only consider edge probabilities in the graphon at points  $(x, y) \in \zeta \times \zeta$ , we need not consider the whole graphon  $\mathcal{W}$  but only the probability matrix  $\mathbf{T} \in [0, 1]^{N \times N}$  that contains the edge probabilities at the sampled points. The graphs  $\mathbf{S}^{(k)}$  are then sampled from the same probability matrix  $\mathbf{T}$ , so

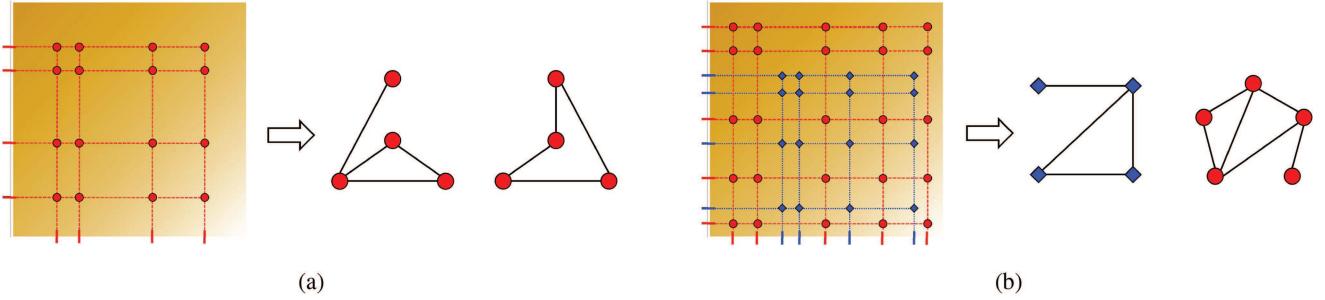


Fig. 1. Schematic depiction of the two problem modalities considered. (a) Multiple graphs sampled from the *same latent point sets* in the *same graphon*. Sampled graphs have not only the same size but also node alignment; see Section IV-A. (b) Multiple graphs sampled from *different latent point sets* in the *same graphon*. Sampled graphs may have different sizes; see Section IV-B.

the graphs all must have the same size, that is,  $N^{(k)} = N$  for all  $k \in \{1, 2, \dots, K\}$ .

Given  $\mathbf{T}$ , the log-likelihood of a graph  $\mathbf{S}^{(k)}$  is

$$\log \Pr \left[ \mathbf{S}^{(k)} | \mathbf{T} \right] = \sum_{i < j} S_{ij}^{(k)} \log (T_{ij}) + \left(1 - S_{ij}^{(k)}\right) \log (1 - T_{ij}),$$

where we have leveraged the fact that, given  $\mathbf{T}$ , edges are drawn independently in our graph model [cf. (1b)]. Furthermore, we can estimate each edge probability  $T_{ij}$  by the sample mean of the graph edges. Thus, we estimate the probability matrix  $\mathbf{T}$  as  $\frac{1}{K} \sum_{k=1}^K \mathbf{S}^{(k)}$ .

Recalling the notation for  $\mathbf{X}$  and  $\mathbf{S}$  from Problem 1, consider a generic optimization problem to estimate multiple networks that we formalize as

$$\min_{\mathbf{S}} \quad f(\mathbf{S}, \mathbf{X}) + L(\mathbf{S}), \quad (2)$$

where the objective function  $f(\mathbf{S}, \mathbf{X})$  estimates graph structures from the observed graph signals, and  $L(\mathbf{S})$  is an additional graph penalty or regularizer; in Section IV-C we provide common examples for these functions. To solve our problem at hand, we propose to append the generic formulation in (2) with a negative log-likelihood penalty to obtain

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{T}} \quad & f(\mathbf{S}, \mathbf{X}) + L(\mathbf{S}) - \sum_{k=1}^K \log \Pr \left[ \mathbf{S}^{(k)} | \mathbf{T} \right] \\ \text{s.to} \quad & \mathbf{S}^{(k)} \in \mathcal{S}_A, \quad \mathbf{T} = \frac{1}{K} \sum_{k=1}^K \mathbf{S}^{(k)}, \end{aligned} \quad (3)$$

where we jointly estimate the graphs and their shared generative probability matrix  $\mathbf{T}$ . The new term promoting maximum likelihood encourages edges to be similar based on shared probabilities in aligned edges. The estimation of the probability matrix entries is included as the sample mean of the edges in the graphs. The set  $\mathcal{S}_A$  enforces valid binary adjacency matrices, that is,

$$\mathcal{S}_A = \left\{ \mathbf{S}^{(k)} : \mathbf{S}^{(k)} = (\mathbf{S}^{(k)})^\top, S_{ii}^{(k)} = 0, S_{ij}^{(k)} \in \{0, 1\} \right\},$$

where we consider undirected graphs without self-loops and edges that are unweighted.

As mentioned in the problem statement, the assumption  $\zeta^{(k)} = \zeta$  for all  $k \in \{1, 2, \dots, K\}$  is equivalent to node alignment for all graphs. We relate (3) to the task of estimating functional networks among the same brain regions of one subject under a set of discrete stimuli, or observing climate variables among the same geographical regions over several time instances.

### B. Graphs and Graphon Estimation

We now consider the case where each graph is sampled from different latent point sets and graphs may possibly have different sizes, i.e.,  $\zeta^{(k)} \neq \zeta^{(k')}$  and  $N^{(k)} \neq N^{(k')}$  for  $k \neq k'$ ; see Fig. 1(b). Therefore, each graph is sampled from a potentially different probability matrix  $\mathbf{T}^{(k)}$ , which is the value of the graphon at the points  $(x, y) \in \zeta^{(k)} \times \zeta^{(k)}$ . For this case, let the set of probability matrices be represented by  $\mathbf{T} = \{\mathbf{T}^{(k)}\}_{k=1}^K$ . The probability matrices provide estimates of the graphon at the known latent point pairs, and each graph provides information about the value of its respective probability matrix.

We again build our joint network and graphon inference framework using (2) as our starting point. We add a new set of terms to encourage maximum likelihood of the graph structures and the graphon model, along with a penalty to incorporate prior graphon information. Under this setting, we present a general optimization framework to jointly estimate the graphs, the probability matrices, and the graphon as

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{T}, \mathcal{W}} \quad & f(\mathbf{S}, \mathbf{X}) + L(\mathbf{S}) - \sum_{k=1}^K \log \Pr \left[ \mathbf{S}^{(k)} | \mathbf{T}^{(k)} \right] + g(\mathcal{W}) \\ \text{s.to} \quad & \mathbf{S}^{(k)} \in \mathcal{S}_A, \\ & \mathbf{T}^{(k)} = h(\mathbf{S}^{(k)}), \quad T_{ij}^{(k)} = \mathcal{W} \left( \zeta_i^{(k)}, \zeta_j^{(k)} \right), \\ & \mathcal{W} : [0, 1]^2 \rightarrow [0, 1], \quad \mathcal{W}(x, y) = \mathcal{W}(y, x), \end{aligned} \quad (4)$$

where we include the same negative log-likelihood term as in (3), but each graph is associated with a different probability matrix  $\mathbf{T}^{(k)}$ . The function  $h(\mathbf{S}^{(k)})$  is a probability matrix estimation method that takes an adjacency matrix  $\mathbf{S}^{(k)}$  as input, such as network histogram or SBM approximations [61], [64]. The third constraint fits the graphon  $\mathcal{W}$  at the known latent point pairs to

the values of the probability matrices  $\mathbf{T}^{(k)}$ , and the regularization term  $g(\mathcal{W})$  in the objective imposes a prior on the overall graphon structure. For example, we may apply a thin-plate spline term [75] to estimate a smooth graphon assuming that points  $\zeta^{(k)}$  are relatively evenly spaced throughout the interval  $[0,1]$ . Other interpolation methods may be applied for smooth graphon estimation, such as inverse distance weighting [76]. The suggested potential interpolation terms assume graphon smoothness to estimate the remainder of the graphon.

The assumptions in (4) are weaker than those in (3), thus a wider range of applications are available. In the example of estimating brain functional networks, functional connectivity of the same subject may be inferred for different sets of brain regions. Additionally, climate network inference is often based on correlation or mutual inference measures, which decreases with geographic distance [2], [5]. Thus, separating inference of climate networks into multiple networks of subregions and applying (4) may be more practical than estimating a single climate network for a large region, as connectivity is expected to be very low for far apart geographical locations.

### C. Examples for Network Inference Methods

Up to this point, we have been considering a generic network inference problem in (2). Both the formulations in (3) and (4) are applicable to existing network inference methods through specific choices of functions  $f(\mathbf{S}, \mathbf{X})$  and  $L(\mathbf{S})$ . Assumptions required for each signal model are explored in Section IV-D.

Consider examples for the function  $f(\mathbf{S}, \mathbf{X})$  that relate the observed graph signals to the structure of the graphs. Graph signals may be assumed to be smooth over their respective graphs [14], and we apply the penalty

$$f(\mathbf{S}, \mathbf{X}) = \sum_{k=1}^K \|\mathbf{S}^{(k)} \circ \mathbf{Z}^{(k)}\|_1, \quad (5)$$

where  $\mathbf{Z}_{ij}^{(k)} = \|\mathbf{X}_i^{(k)} - \mathbf{X}_j^{(k)}\|^2$  as in [14]. Alternatively, we may have graph signals that are the diffusion of noise through a graph filter [10], [15]. In this case, we have stationary graph signals, where the signal covariance  $\mathbf{C}$  commutes with the adjacency matrix  $\mathbf{S}$ . Defining sample covariance matrices as  $\mathbf{C}^{(k)} = \frac{1}{r_k} \mathbf{X}^{(k)} (\mathbf{X}^{(k)})^\top \in \mathbb{R}^{N^{(k)} \times N^{(k)}}$ , we can write

$$f(\mathbf{S}, \mathbf{X}) = \sum_{k=1}^K \|\mathbf{S}^{(k)} \mathbf{C}^{(k)} - \mathbf{C}^{(k)} \mathbf{S}^{(k)}\|_F^2. \quad (6)$$

In many applications, the graphs of interest are sparse, so it is common to apply a sparsity constraint for each graph [11], [15]. We may apply this with the penalty function  $L(\mathbf{S})$  as

$$L(\mathbf{S}) = \sum_{k=1}^K \|\text{vec}(\mathbf{S}^{(k)})\|_1. \quad (7)$$

Note that while the graphon model results in dense graphs as the graph size grows, individually sampled graphs of finite size may be sparse in the sense that the adjacency matrices may contain many zero entries. For example, an Erdős-Rényi graph with low edge probability will have a small ratio of edges to pairs of

nodes. In this case, inclusion of a sparsity promoting penalty would improve network recovery performance.

If, instead of separately inferring each graph, we wish to promote similar sparsity patterns, we may encourage edge similarity between graphs [29], [34] as

$$L(\mathbf{S}) = \sum_{k < k'} \|\text{vec}(\mathbf{S}^{(k)} - \mathbf{S}^{(k')})\|_1, \quad (8)$$

which requires graphs that are not only the same size, but are also on the same node set. Thus, the regularizer in (8) is applicable to our formulation in (3) but not to the one in (4).

Combinations of the described examples for  $f(\mathbf{S}, \mathbf{X})$  and  $L(\mathbf{S})$  are common in existing works. For instance, graph signal stationarity in (6) and sparsity penalties for each graph via (7) are applied in [15]. Moreover, joint inference is performed in [34] by combining (6) and (8).

### D. Assumptions

The proposed formulations in (3) and (4) offer flexible methods for inference of multiple networks, where we may enforce characteristics in the inferred networks based on prior knowledge. What follows is an organization of the assumptions required for our approach.

**Problem assumptions.** Section III introduces the assumptions about our considered problem. Assumption (AS1) requires that networks share the same generating graphon. Typically, joint inference of multiple networks seeks similar edge values by explicitly encouraging edges of the same node pairs to be as close as possible (see Section II-D), while our stochastic approach is less stringent. For each node pair that occurs in multiple networks, each network shares the *likelihood* that an edge will connect the node pair. Furthermore, we allow nodes to belong to different node sets across networks. Given node values in a latent space, edge probabilities for two node pairs are similar if their pairs of latent points are close to each other. This is conceptually similar to the case of shared node pairs, but now the latent space dictates similarity in stochastic behavior for all edges.

In assumption (AS2), we require knowledge of node assignments in a latent space, which we may incorporate via nodal features whose interactions dictate the presence of edges. For example, if two nodes belonging to the same class are expected to be connected, and the features of each node indicate its class assignment, then we may apply nodal features to inform the points in the latent space. We demonstrate an example of using node features for latent point assignment in Section VII-B, where it is more likely for two senators (nodes) to be connected if their political parties (classes) are the same.

**Graph signal model.** For the penalty (5), we assume that graph signals are smooth on their respective graphs, where we expect well-connected nodes to have similar signal values [14]. In particular, we assume that data on the graphs lie on a smooth manifold, where nodes denote points in the manifold space and edges reflect distances between points. We apply penalty (6) for stationary graph signals, where we assume that graph signals are diffusions of arbitrary input through graph filters. If we

assume *linear, shift-invariant* graph filters and *white noise* input signals, then the graph signal covariances and the network structures become spectrally related. In particular, eigenvectors are equivalent, thus the covariance matrices and adjacency matrices commute.

**Network and graphon structure.** We often wish to obtain the most parsimonious network representations for interpretability of connectivity and mitigation of downstream computation. It is common to encourage sparsity in inferred networks via the  $\ell_1$ -norm or Frobenius norm, hence the penalty (7). The joint network inference trademark of encouraging edge similarity in (8) is effective, but it requires that nodes lie on the same node set, an assumption which we relax in (4). When all networks share the same node set, combining the edge-wise similarity penalty (8) and shared edge probabilities in our proposed formulation (3) is an appropriate course of action.

Finally, the choice of  $g(\mathcal{W})$  in (4) allows us to incorporate desirable characteristics of the graphon. For example, we assume smoothness in the underlying graphon, where two node pairs with similar latent point pairs are expected to have similar edge probabilities.

## V. ALGORITHM DEVELOPMENT

We address both (3) and (4) via an alternating direction method of multipliers (ADMM) algorithm [77]. ADMM is an attractive approach as it allows decoupling terms that cannot easily be optimized jointly and handling nonconvex constraints such as our unweighted graph condition. We terminate optimization after reaching the criteria presented in [77, Section 3.3.1]. We solve (4) by alternately optimizing the adjacency matrices, the probability matrices, and the shared graphon. To avoid redundancy, we delay presentation of our solution to (3), as it is a simpler version of (4). Indeed, if we let the penalty  $g(\mathcal{W}) = 0$  and the constraint  $\mathbf{T}^{(k)} = \mathbf{T} = \frac{1}{K} \sum_{k=1}^K \mathbf{S}^{(k)}$  for all graphs  $k \in \{1, 2, \dots, K\}$ , then the problem (4) reduces to the formulation in (3).

The general formulation (4) is difficult to solve due to the continuous graphon penalties and constraints. However, we can relax it to a computationally feasible problem by estimating a *discretized graphon* [61], [66], [78]. We replace the graphon  $\mathcal{W}$  with a discrete matrix counterpart  $\mathbf{W} \in [0, 1]^{G \times G}$ . Selection of the size  $G$  requires a tradeoff between the fineness of the grid to accurately estimate the graphon  $\mathcal{W}$  and the coarseness of the matrix  $\mathbf{W}$  to minimize computational complexity [78]. In this paper, we let the size  $G$  be dependent on the network sizes, e.g.,  $D + \sum_k C N^{(k)}$  for integers  $C \geq 1$  and  $D \geq 0$ , where values of  $C$  and  $D$  can be set as large as computational ability allows. We empirically observe that this choice of discretization is adequate for our proposed joint network and graphon estimation. We leave investigating other choices of graphon discretization as future work [78], [79].

Our proposed relaxation is as follows

$$\min_{\mathbf{S}, \mathbf{T}, \mathbf{W}} f(\mathbf{S}, \mathbf{X}) + L(\mathbf{S}) - \sum_{k=1}^K \log \Pr \left[ \mathbf{S}^{(k)} | \mathbf{T}^{(k)} \right] + \bar{g}(\mathbf{W})$$

s.to  $\mathbf{S}^{(k)} \in \mathcal{S}_A$ ,  $\mathbf{W} \in \mathcal{S}_{\mathcal{W}}$ ,

$$\begin{aligned} \|\mathbf{T}^{(k)} - h(\mathbf{S}^{(k)})\|_F^2 &\leq \epsilon_1^{(k)}, \\ \|\mathbf{T}^{(k)} - \mathbf{W}_{\mathbf{z}^{(k)} \mathbf{z}^{(k)}}\|_F^2 &\leq \epsilon_2^{(k)}, \end{aligned} \quad (9)$$

where  $\mathbf{W}_{\mathbf{z}^{(k)} \mathbf{z}^{(k)}} \in [0, 1]^{N^{(k)} \times N^{(k)}}$  is the submatrix of  $\mathbf{W}$  whose entries consist of the graphon  $\mathcal{W}$  values at points  $(x, y) \in \zeta^{(k)} \times \zeta^{(k)}$  for each graph, and  $\bar{g}(\mathbf{W})$  acts as the discretization of the operation  $g(\mathcal{W})$ . Furthermore, the set  $\mathcal{S}_{\mathcal{W}}$  defines valid discretized graphon matrices as

$$\mathcal{S}_{\mathcal{W}} = \{ \mathbf{W} : \mathbf{W} = \mathbf{W}^T, W_{ij} \in [0, 1] \}.$$

Since we only consider graphs without self-loops, diagonal values of sampled adjacency matrices are ignored [cf. (1b)].

Note that we also relax the equality constraints fitting each probability matrix  $\mathbf{T}^{(k)}$  to each graphon submatrix  $\mathbf{W}_{\mathbf{z}^{(k)} \mathbf{z}^{(k)}}$  and probability matrix estimate  $h(\mathbf{S}^{(k)})$ . We expect upper bounds  $\epsilon_1^{(k)}$  and  $\epsilon_2^{(k)}$  to depend on the number of graphs  $K$  and the graph sizes  $N^{(k)}$ . Greater values of  $K$  and  $N^{(k)}$  increase the number of sampled points in the graphon space, resulting in more precise graphon estimation. Additionally, as  $N^{(k)}$  grows, the probability matrix  $\mathbf{T}^{(k)}$  approaches the underlying graphon  $\mathcal{W}$ , converging to a continuous graphon approximation.

We highlight three major benefits of our formulation: (i) graphs of different sizes can be inferred, (ii) explicit knowledge of the graphon  $\mathcal{W}$  is not needed, and (iii) the relaxed problem (9) is well suited to alternating minimization with optimizing each variable  $\mathbf{S}$ ,  $\mathbf{T}$ , and  $\mathbf{W}$  while fixing the others. In the sequel, we consider special cases of the functions  $f(\mathbf{S}, \mathbf{X})$ ,  $L(\mathbf{S})$ ,  $h(\mathbf{S}^{(k)})$ , and  $g(\mathcal{W})$  to provide a concrete example of our proposed problem and demonstrate its implementation in a common GSP scenario.

### A. Stationary Graph Signals and Smooth Graphon

To demonstrate the implementation of our multiple graph learning algorithm, we present a special case under specific assumptions. We let  $f(\mathbf{S}, \mathbf{X})$  take the form of (6) and let  $L(\mathbf{S}) = 0$ . We choose  $h(\mathbf{S}^{(k)})$  as a network histogram method to estimate the probability matrices  $\mathbf{T}^{(k)}$  [61], [64]. In particular, the probability matrix corresponding to the adjacency matrix  $\mathbf{S}^{(k)}$  can be estimated via SBM approximation [61] as

$$\hat{\mathbf{T}}^{(k)} = h(\mathbf{S}^{(k)}) = \mathbf{F}^{(k)} \mathbf{S}^{(k)} \mathbf{F}^{(k)},$$

where  $\mathbf{F}^{(k)} = \frac{f}{N^{(k)} - f} (\mathbf{I}_{N^{(k)}} / f \otimes (\mathbf{1}_f \mathbf{1}_f^\top - \mathbf{I}_f))$  computes the empirical edge probability of adjacency matrix blocks with size  $f > 0$ . We point out that knowledge of the graphon latent sample points obviates the need for sorting the adjacency matrices by degrees before computing the approximate SBM [61]. Finally, we let  $g(\mathcal{W})$  be a thin-plate spline term [75]

$$g(\mathcal{W}) = \int_0^1 \int_0^1 \left( \frac{\partial^2 \mathcal{W}}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 \mathcal{W}}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 \mathcal{W}}{\partial y^2} \right)^2 dx dy$$

which we then discretize for implementing  $\bar{g}(\mathbf{W})$ . First, we introduce the difference matrices  $\mathbf{D}_1 \in \mathbb{R}^{G \times G-1}$  and  $\mathbf{D}_2 \in$

$\mathbb{R}^{G \times G-2}$  such that

$$[\mathbf{D}_1]_{ij} = \begin{cases} -1, & i-j=0 \\ 1, & i-j=1 \\ 0, & \text{otherwise} \end{cases}$$

and

$$[\mathbf{D}_2]_{ij} = \begin{cases} 1, & i-j \in \{0, 2\} \\ -2, & i-j=1 \\ 0, & \text{otherwise} \end{cases},$$

and we discretize  $g(\mathcal{W})$  as

$$\bar{g}(\mathbf{W}) = \|\mathbf{D}_2^\top \mathbf{W}\|_F^2 + \|\mathbf{D}_1^\top \mathbf{W} \mathbf{D}_1\|_F^2 + \|\mathbf{W} \mathbf{D}_2\|_F^2.$$

Under these selections, we assume that graph signals are stationary on their respective adjacency matrices; see the discussion of (6) and Section IV-D. Our choices of  $(\mathbf{S}^{(k)})$  and  $\bar{g}(\mathbf{W})$  assume that the underlying graphon  $\mathcal{W}$  is smooth [61], [75].

Note that all matrices are symmetric with irrelevant diagonal values. Thus, we introduce vectors containing the matrix lower triangular entries. First, let  $L^{(k)} = N^{(k)}(N^{(k)}-1)/2$ ,  $J = G(G+1)/2$ , and  $L_K = \sum_{k=1}^K L^{(k)}$ . We define  $\mathbf{s} = [(\mathbf{S}^{(1)})_{\mathcal{L}}^\top, \dots, (\mathbf{S}^{(K)})_{\mathcal{L}}^\top]^\top \in \mathbb{R}^{L_K}$ , collecting lower triangles of matrices in the tuple  $\mathbf{S}$  in a column vector. The vector  $\mathbf{t} \in \mathbb{R}^{L_K}$  is defined similarly. We also let  $\mathbf{w} = [\mathbf{W}_{\mathcal{D}}^\top \mathbf{W}_{\mathcal{L}}^\top]^\top$  such that  $\mathbf{w} \in \mathbb{R}^J$  contains all entries of  $\text{vec}(\mathbf{W})$  corresponding to the indices  $\mathcal{D} \cup \mathcal{L}$ .

We introduce the matrix  $\mathbf{M}$  such that  $\|\mathbf{Ms}\|_2^2 = f(\mathbf{S}, \mathbf{X})$  [34], along with  $\Psi$  for applying the network histogram method to  $\mathbf{s}$ . We define

$$\begin{aligned} \mathbf{M}^{(k)} &= [\mathbf{C}^{(k)} \oplus -\mathbf{C}^{(k)}]_{\mathcal{L}}^\top + [\mathbf{C}^{(k)} \oplus -\mathbf{C}^{(k)}]_{\mathcal{U}}^\top, \\ \Psi^{(k)} &= [\mathbf{F}^{(k)} \otimes \mathbf{F}^{(k)}]_{\mathcal{L}}^\top + [\mathbf{F}^{(k)} \otimes \mathbf{F}^{(k)}]_{\mathcal{U}}^\top. \end{aligned}$$

We also introduce  $\Sigma^{(k)} = [\Sigma_{\mathcal{D}}^{(k)} \Sigma_{\mathcal{L}}^{(k)}]$  for selecting graphon indices from  $\mathbf{w}$  by defining

$$\begin{aligned} \Sigma_{\mathcal{D}}^{(k)} &= [\mathbf{I}_{\mathbf{z}^{(k)}} \otimes \mathbf{I}_{\mathbf{z}^{(k)}}]_{\mathcal{D}}^\top, \\ \Sigma_{\mathcal{L}}^{(k)} &= [\mathbf{I}_{\mathbf{z}^{(k)}} \otimes \mathbf{I}_{\mathbf{z}^{(k)}}]_{\mathcal{L}}^\top + [\mathbf{I}_{\mathbf{z}^{(k)}} \otimes \mathbf{I}_{\mathbf{z}^{(k)}}]_{\mathcal{U}}^\top, \end{aligned}$$

where  $\mathbf{I}_{\mathbf{z}^{(k)}}$  denotes a subset of rows of the identity matrix  $\mathbf{I}_G$  indexed by  $\mathbf{z}^{(k)}$ . We can then introduce the block matrices  $\mathbf{M} = \text{blockdiag}(\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \dots, \mathbf{M}^{(K)})$ ,  $\Sigma = [(\Sigma^{(1)})^\top (\Sigma^{(2)})^\top \dots (\Sigma^{(K)})^\top]^\top$ , and  $\Psi = \text{blockdiag}(\Psi^{(1)}, \Psi^{(2)}, \dots, \Psi^{(K)})$ . We let

$$\Phi = \begin{bmatrix} \Phi_{\mathcal{D},1} & \Phi_{\mathcal{L},1} \\ \Phi_{\mathcal{D},2} & \Phi_{\mathcal{L},2} \end{bmatrix}$$

such that  $\|\Phi \mathbf{w}\|_2^2 = \bar{g}(\mathbf{W})$ , where

$$\begin{aligned} \Phi_{\mathcal{D},1} &= [\mathbf{D}_2 \otimes \mathbf{I}_G]_{\mathcal{D}}^\top, \\ \Phi_{\mathcal{L},1} &= [\mathbf{D}_2 \otimes \mathbf{I}_G]_{\mathcal{L}}^\top + [\mathbf{D}_2 \otimes \mathbf{I}_G]_{\mathcal{U}}^\top, \\ \Phi_{\mathcal{D},2} &= [(\mathbf{D}_1 \otimes \mathbf{D}_1)]_{\mathcal{D}}^\top, \\ \Phi_{\mathcal{L},2} &= [(\mathbf{D}_1 \otimes \mathbf{D}_1)]_{\mathcal{L}}^\top + [(\mathbf{D}_1 \otimes \mathbf{D}_1)]_{\mathcal{U}}^\top. \end{aligned}$$

Finally, we define the following function

$$\Gamma(\mathbf{s}, \mathbf{t}) = - \sum_{i=1}^{L_K} [s_i \log(t_i) + (1-s_i) \log(1-t_i)]$$

to represent the log-likelihood term.

From the preceding definitions, we can rewrite the problem (9) to eliminate constraints and minimize the number of variables to be optimized. This simplification is shown in the following vectorized problem

$$\begin{aligned} \min_{\mathbf{s}, \mathbf{t}, \mathbf{w}} \quad & \frac{\alpha}{2} \|\mathbf{Ms}\|_2^2 + \Gamma(\mathbf{s}, \mathbf{t}) + \frac{\beta}{2} \|\Phi \mathbf{w}\|_2^2 \\ \text{s.to} \quad & s_i \in \{0, 1\}, \quad w_i \in [0, 1], \\ & \|\mathbf{t} - \Psi \mathbf{s}\|_2^2 \leq \epsilon_1, \quad \|\mathbf{t} - \Sigma \mathbf{w}\|_2^2 \leq \epsilon_2 \end{aligned} \quad (13)$$

with tuning parameters  $\alpha > 0$  and  $\beta > 0$  to control graph signal stationarity and graphon smoothness.

Developing an ADMM algorithm with guaranteed convergence requires reformulating (13) not only to incorporate dual variables and parameters but also to account for the constraints on the entries in  $\mathbf{s}$  and  $\mathbf{w}$ . We introduce the auxiliary variables  $\mathbf{p} \in \mathbb{R}^{L_K}$  and  $\mathbf{v} \in \mathbb{R}^J$  and expand the problem as follows

$$\begin{aligned} \min_{\mathbf{s}, \mathbf{t}, \mathbf{w}, \mathbf{p}, \mathbf{v}} \quad & \frac{\alpha}{2} \|\mathbf{Ms}\|_2^2 + \Gamma(\mathbf{s}, \mathbf{t}) + \frac{\beta}{2} \|\Phi \mathbf{w}\|_2^2 \\ \text{s.to} \quad & \mathbf{s} = \mathbf{p}, \quad \mathbf{w} = \mathbf{v}, \\ & p_i \in \{0, 1\}, \quad v_i \in [0, 1], \\ & \|\mathbf{t} - \Psi \mathbf{s}\|_2^2 \leq \epsilon_1, \quad \|\mathbf{t} - \Sigma \mathbf{w}\|_2^2 \leq \epsilon_2. \end{aligned}$$

The augmented Lagrangian function then takes the form [77]

$$\begin{aligned} \mathfrak{L}_\rho(\mathbf{s}, \mathbf{p}, \mathbf{t}, \mathbf{w}, \mathbf{v}, \mathbf{u}_1, \mathbf{u}_2) = & \frac{\alpha}{2} \|\mathbf{Ms}\|_2^2 + \Gamma(\mathbf{s}, \mathbf{t}) + \frac{\beta}{2} \|\Phi \mathbf{w}\|_2^2 \\ & + \mathbb{I}\{\mathbf{p}_i \in \{0, 1\} \forall i\} + \mathbb{I}\{\mathbf{v}_i \in [0, 1] \forall i\} \\ & + \rho_1 \langle \mathbf{u}_1, \mathbf{s} - \mathbf{p} \rangle + \frac{\rho_1}{2} \|\mathbf{s} - \mathbf{p}\|_2^2 \\ & + \rho_2 \langle \mathbf{u}_2, \mathbf{w} - \mathbf{v} \rangle + \frac{\rho_2}{2} \|\mathbf{w} - \mathbf{v}\|_2^2 \\ & + \frac{\lambda_1}{2} \|\mathbf{t} - \Psi \mathbf{s}\|_2^2 + \frac{\lambda_2}{2} \|\mathbf{t} - \Sigma \mathbf{w}\|_2^2, \end{aligned} \quad (14)$$

where  $\mathbb{I}\{\cdot\}$  is the indicator function that takes the value 0 if the argument is true and infinity otherwise. The primal variables consist of  $\mathbf{s}$ ,  $\mathbf{p}$ ,  $\mathbf{t}$ ,  $\mathbf{w}$ , and  $\mathbf{v}$ , the dual variables  $\mathbf{u}_1$  and  $\mathbf{u}_2$ , and the dual parameters  $\rho_1$  and  $\rho_2$ . Tuning parameters  $\lambda_1$  and  $\lambda_2$  determine the strength of the relationships among the adjacency matrices, the graphon, and the probability matrices. We also define the function  $\Pi_{\mathcal{C}}(\cdot)$  as the projection of the argument onto the set  $\mathcal{C}$ .

Applying ADMM to (14) results in the following update steps

$$\mathbf{s}^{j+1} = \underset{\mathbf{s}}{\operatorname{argmin}} \mathfrak{L}_\rho(\mathbf{s}, \mathbf{p}^j, \mathbf{t}^j, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j) \quad (15a)$$

$$\mathbf{p}^{j+1} = \Pi_{\{0,1\}}(\mathbf{s}^{j+1} + \mathbf{u}_1^j) \quad (15b)$$

$$\mathbf{t}^{j+1} = \underset{\mathbf{t}}{\operatorname{argmin}} \mathfrak{L}_\rho(\mathbf{s}^{j+1}, \mathbf{p}^{j+1}, \mathbf{t}, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j) \quad (15c)$$

$$\mathbf{w}^{j+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{s}^{j+1}, \mathbf{p}^{j+1}, \mathbf{t}^{j+1}, \mathbf{w}, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j) \quad (15d)$$

$$\mathbf{v}^{j+1} = \Pi_{[0,1]}(\mathbf{w}^{j+1} + \mathbf{u}_2^j) \quad (15e)$$

$$\mathbf{u}_1^{j+1} = \mathbf{u}_1^j + \mathbf{s}^{j+1} - \mathbf{p}^{j+1} \quad (15f)$$

$$\mathbf{u}_2^{j+1} = \mathbf{u}_2^j + \mathbf{w}^{j+1} - \mathbf{v}^{j+1} \quad (15g)$$

What follows is a discussion of non-trivial update steps.

**Optimization of Graphs.** Adjacency matrices are initialized via general graph estimation in (2). We update the graphs  $\mathbf{s}$  by solving the problem (15a), resulting in the closed form solution

$$\begin{aligned} & (\alpha \mathbf{M}^\top \mathbf{M} + \rho_1 \mathbf{I}_{L_K} + \lambda_1 \boldsymbol{\Psi}^\top \boldsymbol{\Psi}) \mathbf{s} \\ &= \boldsymbol{\gamma}(\mathbf{t}^j) + \rho_1(\mathbf{p}^j - \mathbf{u}_1^j) + \lambda_1 \boldsymbol{\Psi}^\top \mathbf{t}^j \end{aligned}$$

where we let  $[\boldsymbol{\gamma}(\mathbf{t})]_i = \log(t_i) - \log(1 - t_i)$ .

**Optimization of Probability Matrices.** We update the probability matrix vector  $\mathbf{t}$  by solving

$$\min_{\mathbf{t}} \mathbf{\Gamma}(\mathbf{s}, \mathbf{t}) + \frac{\lambda_1}{2} \|\mathbf{t} - \boldsymbol{\Psi} \mathbf{s}^{j+1}\|_2^2 + \frac{\lambda_2}{2} \|\mathbf{t} - \boldsymbol{\Phi} \mathbf{w}^j\|_2^2.$$

The subproblem for updating  $\mathbf{t}$  is separable by entries of  $\mathbf{t}$ , thus we can parallelize computation by solving the equivalent problem

$$\begin{aligned} & \min_{t_i} - (s_i^{j+1} \log(t_i) + (1 - s_i^{j+1}) \log(1 - t_i)) \\ &+ \frac{c}{2} (t_i - d)^2 \end{aligned} \quad (16)$$

where  $c = \lambda_1 + \lambda_2$  and  $d = \frac{\lambda_1}{c} \boldsymbol{\Psi}_i \mathbf{s}^{j+1} + \frac{\lambda_2}{c} \boldsymbol{\Phi}_i \mathbf{w}^j$ , obtained by completing the square of the original subproblem. As the function  $s_i \log(t_i) + (1 - s_i) \log(1 - t_i)$  does not have an easily computed proximal operator, we obtain the update for  $\mathbf{t}$  by solving (16) via proximal gradient descent (PG) [80], and we observe that PG converges quickly in practice.

**Optimization of Graphon.** The solution to the graphon subproblem (15d) is also a closed-form expression,

$$(\beta \boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} + \rho_2 \mathbf{I}_J + \lambda_2 \boldsymbol{\Phi}^\top \boldsymbol{\Phi}) \mathbf{w} = \rho_2(\mathbf{v}^j - \mathbf{u}_2^j) + \lambda_2 \boldsymbol{\Phi}^\top \mathbf{t}^{j+1}.$$

We thus obtain a *convergent result* for solving (14). The main result is shown in the following theorem.

**Theorem 1:** When ADMM with update steps (15a)–(15g) is applied to the joint network and graphon inference optimization problem (14), if the underlying generating graphon  $\mathcal{W}$  is bounded away from 0 and 1, i.e., there exists some  $\epsilon > 0$  such that  $\mathcal{W}(x, y) \in [\epsilon, 1 - \epsilon]$  for all  $x, y \in [0, 1]$ , then for large enough parameters  $\rho_1, \rho_2, \lambda_1$ , and  $\lambda_2$ , the resulting sequence  $(\mathbf{s}^j, \mathbf{p}^j, \mathbf{t}^j, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j)$  has at least one limit point  $(\mathbf{s}^*, \mathbf{p}^*, \mathbf{t}^*, \mathbf{w}^*, \mathbf{v}^*, \mathbf{u}_1^*, \mathbf{u}_2^*)$ , and each is also a stationary point, that is,  $0 \in \partial \mathcal{L}_\rho(\mathbf{s}^*, \mathbf{p}^*, \mathbf{t}^*, \mathbf{w}^*, \mathbf{v}^*, \mathbf{u}_1^*, \mathbf{u}_2^*)$ .

*Proof of Theorem 1:* See Appendix A.

The most complex ADMM steps are the updates for  $\mathbf{s} \in \mathbb{R}^{L_K}$  and  $\mathbf{w} \in \mathbb{R}^J$ . If we assume for simplicity that all graphs have the same number of nodes,  $N^{(k)} = N$  for all  $k \in \{1, 2, \dots, K\}$ , then we have that  $L_K = KN(N - 1)/2$ . We further let the size of the discretized graphon be  $G = CKN + D$  for integers  $C \geq$

1 and  $D \geq 0$ . Thus, each iteration of the update steps (15a)–(15g) has complexity  $\mathcal{O}(J^2) = \mathcal{O}(K^4 N^4)$ , stemming from the graphon vector  $\mathbf{w}$  update.

We may precompute the inverses for updating  $\mathbf{s}$  and  $\mathbf{w}$ , which have complexities  $\mathcal{O}(L_K^3) = \mathcal{O}(K^3 N^6)$  and  $\mathcal{O}(J^3) = \mathcal{O}(K^6 N^6)$ , respectively.

**Remark 1 (Solution to (3)):** When solving the problem in (3), we let  $\mathbf{T}^{(k)} = \mathbf{T}$  for all  $k \in \{1, 2, \dots, K\}$  and ignore graphon penalties and constraints, i.e.,  $\bar{g}(\mathbf{W})$ ,  $\mathbf{W} \in \mathcal{S}_W$ , and  $\|\mathbf{T}^{(k)} - \mathbf{W}_{\mathbf{z}^{(k)} \mathbf{z}^{(k)}}\|_F^2 \leq \epsilon_2^{(k)}$ . In this case, we forgo update steps (15d), (15e), and (15g). The update for  $\mathbf{s}$  remains the same while letting  $\lambda_1 = 0$ .

For the update of  $\mathbf{t}$ , we note that since  $\mathbf{T}^{(k)} = \mathbf{T}$  for all graphs, we have  $L = N(N - 1)/2$  degrees of freedom for (3) instead of the  $L_K = \sum_{k=1}^K L^{(k)}$  of (4). Thus we need only estimate one vector  $\mathbf{t}^{(k)} = \mathbf{t} \in \mathbb{R}^L$  for all graphs. We introduce the matrix  $\mathbf{R} = \frac{1}{K}(\mathbf{1}_K^\top \otimes \mathbf{I}_L)$  such that  $\mathbf{t} = \mathbf{R}\mathbf{s}$ , equivalent to the last constraint in (3). The update step (15c) then becomes

$$\min_{t_i} -K(t_i \log(t_i) + (1 - t_i) \log(1 - t_i)) + \frac{c'}{2} (t_i - d')^2,$$

where  $c' = \lambda_1$  and  $d' = \mathbf{R}_i \mathbf{s}^{j+1}$ . The objective in this subproblem is a difference of convex terms, which can be solved via existing methods such as DCA algorithms [81], [82].

**Remark 2 (Normalized projections):** The Euclidean projections  $\Pi_{\{0,1\}}(\cdot)$  and  $\Pi_{[0,1]}(\cdot)$  applied respectively in (15b) and (15e) guarantee convergence of the ADMM algorithm. However, we empirically observed improved performance by normalized projections. In particular, the argument  $\mathbf{x}$  is first normalized as

$$\bar{\mathbf{x}} = \frac{\mathbf{x} - \min_i x_i}{\max_j(\mathbf{x} - \min_i x_i)},$$

and the normalized  $\bar{\mathbf{x}}$  is then projected to the set  $\mathcal{C}$  as usual as  $\Pi_{\mathcal{C}}(\bar{\mathbf{x}})$ .

## VI. ROBUST NETWORK AND GRAPHON ESTIMATION

In practice, the exact placement of the latent sample points may not be available. For example, electrode location for measuring neural responses may not be precisely comparable across several subjects. For comparisons of conditions in a climate system, regions of observation may not be consistent over time since atmospheric patterns will not necessarily occur in precisely the same geographical locations. Thus, we relax the assumption (AS2) that latent points  $\zeta^{(k)}$  are exactly known, but approximate points  $\bar{\zeta}^{(k)} = \zeta^{(k)} + \omega^{(k)}$  are known, where  $\omega^{(k)}$  is random noise perturbing the true sample points. This assumption was also applied in [83] for graphon estimation from known graphs. We update the problem (4) as

$$\begin{aligned} & \min_{\mathbf{S}, \mathbf{T}, \mathcal{W}, \zeta} f(\mathbf{S}, \mathbf{X}) + L(\mathbf{S}) - \sum_{k=1}^K \log \Pr \left[ \mathbf{S}^{(k)} | \mathbf{T}^{(k)} \right] + g(\mathcal{W}) \\ & \text{s.to} \quad \mathbf{S}^{(k)} \in \mathcal{S}_A, \\ & \quad \mathbf{T}^{(k)} = h(\mathbf{S}^{(k)}), \quad T_{ij}^{(k)} = \mathcal{W} \left( \zeta_i^{(k)}, \zeta_j^{(k)} \right), \\ & \quad \mathcal{W} : [0, 1]^2 \rightarrow [0, 1], \quad \mathcal{W}(x, y) = \mathcal{W}(y, x), \\ & \quad \|\zeta^{(k)} - \bar{\zeta}^{(k)}\|_2^2 \leq \epsilon_3^{(k)}, \end{aligned} \quad (17)$$

where we introduce the final constraint to estimate graphon sample points  $\zeta^{(k)}$  based on the given noisy values  $\bar{\zeta}^{(k)}$  for all graphs  $k \in \{1, 2, \dots, K\}$ .

Implementation is almost identical to that of (4), where we discretize the graphon and perform an ADMM algorithm. The update steps for each variable are the same, with the addition of an update for optimizing the graphon indices  $\mathbf{z}^{(k)}$ . Let the vector  $\mathbf{z}$  be the concatenation of the graphon indices,  $\mathbf{z} = [(\mathbf{z}^{(1)})^\top (\mathbf{z}^{(2)})^\top \dots (\mathbf{z}^{(K)})^\top]^\top$ . The appended update step solves the subproblem

$$\begin{aligned} \mathbf{z}^{j+1} = \underset{\mathbf{z}}{\operatorname{argmin}} \|\mathbf{t}^{j+1} - \bar{\Sigma} \mathbf{w}^{j+1}\|_2^2 \\ \text{s.t. } \bar{\Sigma} = \operatorname{blockdiag}(\Sigma^{(1)}, \Sigma^{(2)}, \dots, \Sigma^{(K)}), \\ \Sigma^{(k)} = [(\mathbf{I}_{\mathbf{z}^{(k)}} \otimes \mathbf{I}_{\mathbf{z}^{(k)}})_{\mathcal{L}}^\top + [(\mathbf{I}_{\mathbf{z}^{(k)}} \otimes \mathbf{I}_{\mathbf{z}^{(k)}})_{\mathcal{U}}^\top]^\top, \\ \mathbf{z}^{(k)} = \bar{\mathbf{z}}^{(k)} + \Delta \mathbf{z}^{(k)}, \\ \Delta z_i^{(k)} \in \{-\eta^{(k)}, -\eta^{(k)} + 1, \dots, \eta^{(k)}\}, \end{aligned} \quad (18)$$

where  $\eta^{(k)}$  is the maximum perturbation of the estimated graphon indices from the given noisy version  $\bar{\mathbf{z}}^{(k)}$ . The value of  $\eta^{(k)}$  depends on the upper bound  $\epsilon_3^{(k)}$  and the size  $G$  of the discretized graphon. The area of the error region of radius  $\epsilon_3^{(k)}$  in the graphon domain  $[0, 1]^2$  dictates the size of the error grid of radius  $\eta^{(k)}$  in the graphon matrix of size  $G \times G$ . We solve this step via a greedy minimization over each entry in  $\mathbf{z}$  using a grid search over the set of perturbations  $\{-\eta^{(k)}, -\eta^{(k)} + 1, \dots, \eta^{(k)}\}$ , where we select the value of  $\Delta z_i^{(k)}$  to minimize the objective. The optimization order is arbitrary; options include sorted or randomized index orders.

## VII. NUMERICAL EXPERIMENTS

We compare the performance of network topology inference methods with and without the augmentations in (3), (4), and (17), denoted by “Mod. 1,” “Mod. 2,” and “Mod. 3,” respectively. For all experiments, we apply the same signal model assumption  $f(\mathbf{S}, \mathbf{X})$  as (6), and we compare separate network inference via sparsity penalties (7) and joint network inference via pairwise difference penalties (8). For synthetic experiments, we sample from the graphon  $\mathcal{W}(x, y) = \frac{1}{2}(x^2 + y^2)$ . The error of estimator  $\hat{\mathbf{S}}$  is calculated as  $\|\mathbf{S} - \hat{\mathbf{S}}\|_F / \|\mathbf{S}\|_F$ , where the true adjacency matrix is given by  $\mathbf{S}$ .<sup>1</sup> Additional results are included in the Supplementary Material.

### A. Synthetic Experiments

**Same node sets.** We consider the case where all graphs are sampled from the same points within the graphon space,  $\zeta^{(k)} = \zeta$ . We estimate  $K = 3$  graphs with  $N = 30$  nodes as we observe an increasing number of signals for sample covariance computation. We present in Fig. 2(a) the comparison of separate and joint network inference methods with the augmentations in (3) and (4), and without either. In both methods, the augmented formulations improve estimation performance significantly. The

pairwise joint penalty (8) enjoys the greatest improvement, as graphs not only possess node alignment required by (8), but they also follow our graph model assumption.

**Node sets of different sizes.** We consider the challenging case where the graphs have different latent point sets of different sizes  $N^{(k)} \neq N^{(k')}$ . Unlike the previous experiment, we cannot apply (3) or (8), so we consider only (6) and (7) with and without the joint graphon estimation from (4). We consider  $K = 3$  graphs for node sets of  $N = 10, 30, 50$  and  $N = 15, 30, 45$  in Fig. 2(b). For both cases, application of joint graphon inference results in consistent improvement, with increasing performance gap for larger number of observed signals.

**Different node sets of same size.** Finally, we observe graphs of the same size and different node sets, i.e.,  $N^{(k)} = N^{(k')}$  but  $\zeta^{(k)} \neq \zeta^{(k')}$  for every pair  $k, k'$ . In this case, (3) is applicable, but the model assumption is incorrect, as it assumes that all graphs are not only sampled from the same graphon, but it is also incorrectly assumed that graphs are sampled from the same probability matrix. We observe in Fig. 2(c) the comparison of the three modalities, separate network inference and joint inference with the augmentations in (3) and (4). While (3) outperforms separate inference, indeed (4) generally exhibits greater improvement as it includes the knowledge of different sample points within the graphon space. Our results demonstrate the value in prior knowledge of sampling locations and accurate estimation of the underlying graphon, as both contribute to improvement of network estimation.

**Noisy latent sample points.** Finally, we observe the performance of the robust formulation in (17) when only a noisy version of the latent sample points  $\zeta^{(k)}$  are available. We infer  $K = 3$  graphs of size  $N = 20$  from the graphon  $W(x, y) = 0.25 + 0.75 \exp\{-\beta(x - 1/2)^2(y - 1/2)^2\}$  for  $\beta > 0$ . We let  $\zeta^{(1)}, \zeta^{(2)} \sim \text{Unif}(0.4, 0.6)$  and  $\zeta^{(3)} \sim \text{Unif}(0.2, 0.4)$ . Latent sample points are perturbed with increasing levels of magnitude, where  $\hat{\zeta}^{(k)} = \zeta^{(k)} + n\omega^{(k)}$  for  $n \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$ . The upper bound  $\epsilon_3^{(k)}$  increases in proportion to the magnitude of the noise as  $\epsilon_3^{(k)} = n$  for all values of  $n$ . Comparisons are shown in Fig. 3. Fig. 3(a) presents comparisons of graph estimation error for our proposed methods, and Fig. 3(b) compares graphon estimation error for formulations (4) and (17). Joint inference of networks and graphons for both the original (4) and robust (17) formulations demonstrate consistent superiority in graph recovery over both separate inference and joint inference of networks and probability matrix (3). Furthermore, the robust inference in (17) demonstrates a general improvement in performance over the original formulation (4). We also observe that graphon estimation is consistently improved when applying the robust joint inference method in Fig. 3(b). Our approach not only demonstrates a viable method for robust graph estimation under noisy latent sample points but also simultaneously presents improved graphon estimation under perturbed prior information.

### B. Senate Networks

Finally, we performed graph estimation with real-world data of U.S. congress roll-call votes [84], and we set up senate vote

<sup>1</sup>Implementations of our method are available at [https://github.com/mn51/jointinf\\_graphon](https://github.com/mn51/jointinf_graphon).

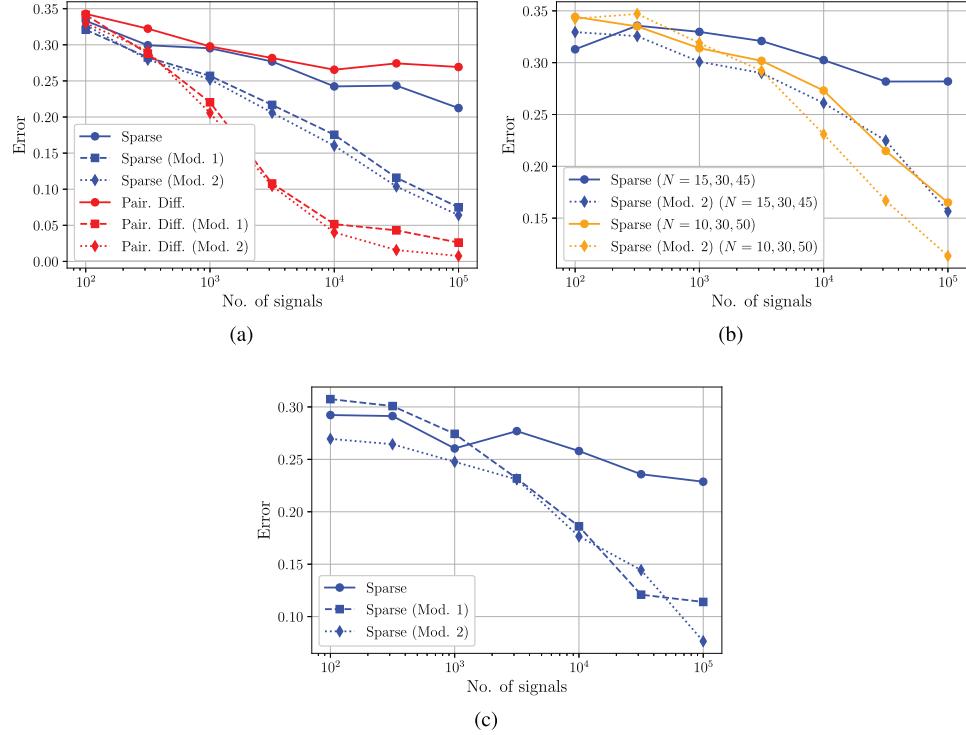


Fig. 2. Performance analysis in synthetic networks. (a) Recovery error for  $K = 3$  graphs sampled from the *same latent point sets* in the same graphon as a function of the number of observed signals. Incorporating the joint estimation of the probability matrix or the graphon both improve estimation performance. (b) Recovery error for  $K = 3$  graphs sampled from *latent point sets of different sizes* in the same graphon as a function of the number of observed signals. Separate inference of graphs is outperformed by including joint estimation of the underlying graphon. (c) Recovery error for  $K = 3$  graphs of the same size  $N = 30$  sampled from *different latent point sets* in the same graphon. Joint estimation of networks and the underlying graphon model drastically outperforms separate inference, and the joint network and graphon estimation (4) demonstrates improvement over joint network and probability matrix estimation (3).

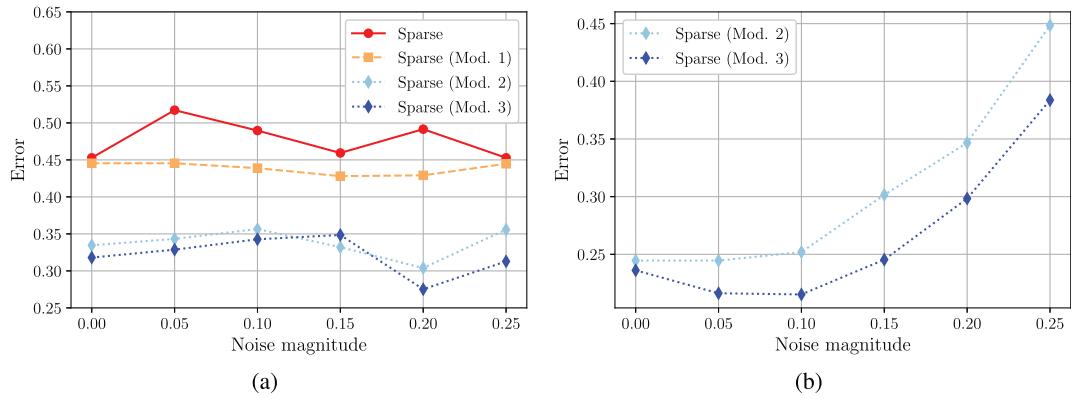


Fig. 3. Performance analysis in synthetic networks under noisy graphon sample information. (a) Recovery error of *networks* as a function of the magnitude of perturbation of graphon latent sample points  $\zeta^{(k)}$  from (1a). Both versions of joint network and graphon inference outperform separate network inference and joint network and probability matrix inference (3) for all noise levels. Robust joint inference (17) exhibits improvement in network recovery compared to (4). (b) Recovery error of *graphon* as a function of the magnitude of perturbation of graphon latent sample points  $\zeta^{(k)}$  from (1a). As the magnitude of perturbation of sample points increases, the robust joint inference (17) demonstrates increasing recovery performance over (4).

signals as in [34]. The number of nodes corresponds to the number of votes (100 senators and 1 President). We let votes represent graph signals, where a node can take the value 1 for *yea*, -1 for *nay*, and 0 for *abstinen*. We observe the 724, 919, and 612 votes of congresses 103, 104, and 105, respectively, and we let the underlying true networks be obtained by separate estimation of each network using all available votes. For the experiment, we consider the same three cases of network estimation as in

the synthetic experiments: (i) networks sampled from the same node set, (ii) networks of different sizes, and (iii) networks of the same size but sampled from different node sets. We estimate subgraphs of the separately inferred true graphs, and senators are chosen for each subgraph analogous to sampling a graphon at selected points.

**Same set of senators.** In the first case, we estimate  $K = 3$  subgraphs sampled from the same points, i.e., networks with

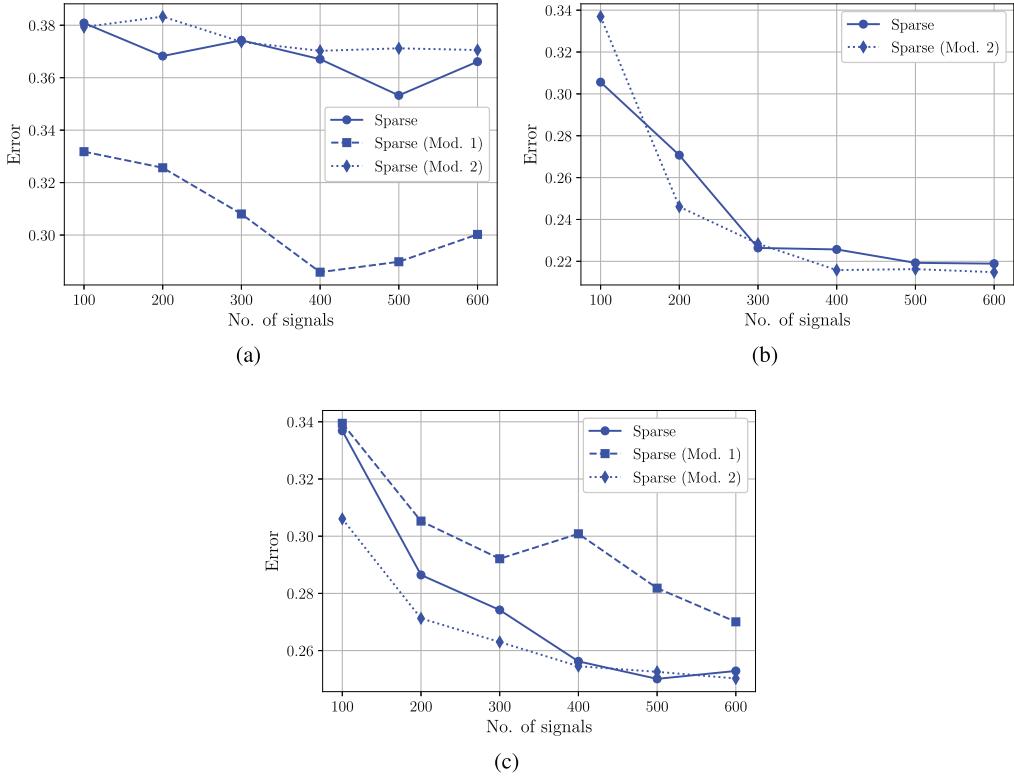


Fig. 4. Performance analysis in senate networks. (a) Recovery error as a function of the number of observed signals for *induced subgraphs* of three senate networks of the same set of senators. Joint network and probability matrix inference (3) outperforms both separate network inference and joint network and graphon inference (4) for all sets of observed signals. (b) Recovery error of *induced subgraphs* of three senate networks of sizes  $N = 15, 30, 45$  as a function of the number of observed signals. Joint network and graphon inference (4) outperforms separate network inference as the set of observed signals grows larger. (c) Recovery error as a function of the number of observed signals for *induced subgraphs* of three senate networks of size  $N = 30$  consisting of different sets of senators. Joint network inference outperforms separate network inference consistently for all sets of observed signals, and joint network and graphon inference (4) demonstrates better performance than joint network and probability matrix inference (3).

nodes  $N = 30$  consisting of the same 30 senators, where we only observe votes of senators corresponding to these subsets of nodes. We show in Fig. 4(a) that (3), which assumes all three node sets consist of the same set of senators, indeed outperforms separate estimation and the augmentation in (4), demonstrating the statistical voting similarities of these chosen senate seats across congresses.

**Sets of senators of different sizes.** For networks of different sizes, we estimate induced subgraphs of sizes  $N = 15, 30, 45$ . In Fig. 4(b) we observe that joint network and graphon estimation from (4) appears to consistently rival or outperform separate inference for larger numbers of observed votes, even though the true networks were estimated separately. Indeed, while separate inference of the induced subgraphs becomes more similar to the true network generation as the number of signals increases, (4) is still able to improve accuracy.

**Different senator sets of same size.** We also revisit the comparison under an incorrect model assumption for (3). We consider  $K = 3$  subgraphs of size  $N = 30$  for all, but the estimated subgraphs consist of different sets of senators. Fig. 4(c) demonstrates that estimation via (4) drastically outperforms that of (3), which assumes the subgraphs consist of the same senators, and furthermore (4) generally outperforms separate inference. These experiments demonstrate that the versatile nonparametric nature of graphons can aid the recovery of real-world graphs,

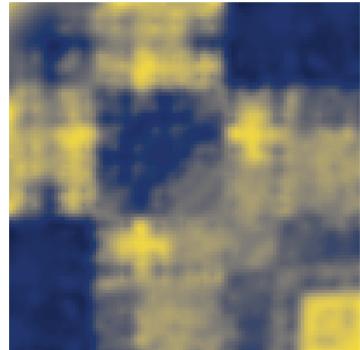


Fig. 5. The graphon estimated jointly with senate networks exhibits two-community node clustering behavior.

even if these graphs have not been explicitly drawn from a graphon model in the first place.

Finally, we include in Fig. 5 the estimated graphon from applying (4) for inferring senate networks. We assume an SBM-like underlying model, where we expect that the nodes (senators) will exhibit clustering behavior into two communities (political parties). The expected behavior is clearly recognizable in the estimated graphon, which validates our claim that the estimated graphon exhibits the general shared structure of a set of real-world networks.

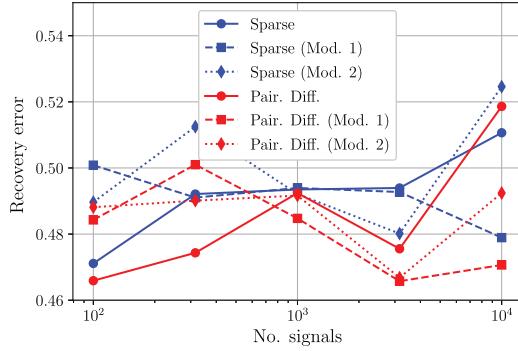


Fig. 6. Performance analysis in brain functional networks. Recovery error as a function of observe time frames for the induced subnetworks of  $K = 3$  different subsets of neurons, each of size  $N = 30$ . Joint network inference assuming an underlying graphon model demonstrates competitive and promising improvement in performance.

### C. Brain Functional Networks

We consider the estimation of neural connectivity using data from the Allen Brain Atlas [85], which consists of recordings of *in vivo* neural responses from 100 neurons of a live mouse being shown visual stimuli, which are designed to stimulate the visual cortex. We let the ground-truth brain functional network be the single estimated network given all 100000 available frames of neural data.

Consider the case where we only observe the activity of multiple subsets of neurons of size  $N = 30$  each, where each subset is measured separately, but we wish to infer the induced subnetworks connecting the observed neurons. We estimate these brain functional subnetworks via the separate network inference method used to generate the ground-truth network, denoted by ‘‘Sparse,’’ which we compare to our proposed joint inference augmentations. Estimation results are shown in Fig. 6, where we observe that for different subsets of neurons, applying our joint inference approaches results in competitive and even superior performances, even when the true underlying network is obtained from the Sparse method.

## VIII. CONCLUSION AND FUTURE WORK

We demonstrated a method to jointly estimate multiple networks under the assumption that they are sampled from the same graphon. To the best of our knowledge, our work is the first to leverage graphons to solve the challenging problem of inferring graphs of different sizes. We also presented a robust method when only noisy information about the latent variables  $\zeta^{(k)}$  is available. We demonstrated that our proposed maximum-likelihood-based method improves network estimation in synthetic and real-world experiments, along with the efficacy of our robust solution with inexact sampling data. In terms of future directions, we plan to consider: i) Alternative random graph models (beyond graphons) that can also generate graphs of different sizes while promoting other common structural characteristics, and ii) Additional relevant tasks on multiple networks that benefit from additional statistical knowledge, i.e., the shared graphon assumption; tasks include graph clustering, dynamic graph change-point detection, or graph node alignment for graphs of different sizes.

## APPENDIX A PROOF OF THEOREM 1

This proof is inspired by [86], with digression mainly due to the function  $g(\mathbf{s}, \mathbf{t}, \mathbf{w})$ , whose Lipschitz smoothness depends on combinations of  $\mathbf{s}$ ,  $\mathbf{t}$ , and  $\mathbf{w}$ , and the order of updates, where the lower semicontinuous indicator functions are not updated first. Additionally, we have that the function  $f(\mathbf{p}, \mathbf{v})$  is not convex with respect to  $\mathbf{p}$ .

*Remark 3 (Boundedness of  $\mathbf{t}$ ):* Note that since the graphon is bounded away from 0 and 1, values of  $\mathcal{W}$  and  $\mathbf{T}^{(k)}$  are restricted to the set  $[\epsilon, 1 - \epsilon]$ . In this proof we assume that  $\mathbf{t}^j \in [\epsilon, 1 - \epsilon]$  for all  $j \in \mathbb{N}$ . This can be easily enforced in the optimization by modifying the log-likelihood penalty  $\hat{\Gamma}(\mathbf{s}, \mathbf{t}) = -\sum_{i=1}^{L_K} [s_i \log(t_i - \epsilon) + (1 - s_i) \log(1 - t_i - \epsilon)]$ , where the resulting modified probability matrix update in (16) remains solvable by proximal gradient descent.

We first introduce the following Lemmas to be used in the remainder of the proof. Proofs of the Lemmas can be found in the Supplementary Material.

*Lemma 1 (Coercivity):* The objective function  $\phi(\mathbf{s}, \mathbf{p}, \mathbf{t}, \mathbf{w}, \mathbf{v})$  is coercive over the feasible set  $\mathcal{F} := \{(\mathbf{s}, \mathbf{p}, \mathbf{t}, \mathbf{w}, \mathbf{v}) \in \text{dom}(\phi) : \mathbf{s} = \mathbf{p}, \mathbf{w} = \mathbf{v}\}$ . Specifically,  $\phi(\mathbf{s}, \mathbf{p}, \mathbf{t}, \mathbf{w}, \mathbf{v}) \rightarrow \infty$  when  $\|(\mathbf{s}, \mathbf{p}, \mathbf{t}, \mathbf{w}, \mathbf{v})\| \rightarrow \infty$ .

*Lemma 2 (Objective regularity):* If  $\mathbf{t} \in [\epsilon, 1 - \epsilon]^{L_K}$  for some  $\epsilon > 0$ , then

a) For positive constants  $G_s^s$ ,  $G_s^t$ ,  $G_w^w$ , and  $G_w^t$ ,

$$\begin{aligned} & \|\nabla_{\mathbf{s}} g(\mathbf{s}, \mathbf{t}, \mathbf{w}) - \nabla_{\mathbf{s}} g(\hat{\mathbf{s}}, \hat{\mathbf{t}}, \hat{\mathbf{w}})\|_2^2 \\ & \leq \lambda_1^2 G_s^s \|\mathbf{s} - \hat{\mathbf{s}}\|_2^2 + \lambda_1^2 G_s^t \|\mathbf{t} - \hat{\mathbf{t}}\|_2^2, \\ & \|\nabla_{\mathbf{t}} g(\mathbf{s}, \mathbf{t}, \mathbf{w}) - \nabla_{\mathbf{t}} g(\mathbf{s}, \mathbf{t}, \hat{\mathbf{w}})\|_2^2 \leq \lambda_2^2 G_w^t \|\mathbf{w} - \hat{\mathbf{w}}\|_2^2, \\ & \|\nabla_{\mathbf{w}} g(\mathbf{s}, \mathbf{t}, \mathbf{w}) - \nabla_{\mathbf{w}} g(\hat{\mathbf{s}}, \hat{\mathbf{t}}, \hat{\mathbf{w}})\|_2^2 \\ & \leq \lambda_2^2 G_w^w \|\mathbf{w} - \hat{\mathbf{w}}\|_2^2 + \lambda_2^2 G_w^t \|\mathbf{t} - \hat{\mathbf{t}}\|_2^2. \end{aligned}$$

b) For positive constants  $H_s$  and  $H_w$ ,

$$\begin{aligned} & \|\nabla_{\mathbf{s}} h(\mathbf{s}, \mathbf{w}) - \nabla_{\mathbf{s}} h(\hat{\mathbf{s}}, \hat{\mathbf{w}})\|_2^2 \leq \alpha^2 H_s \|\mathbf{s} - \hat{\mathbf{s}}\|_2^2, \\ & \|\nabla_{\mathbf{w}} h(\mathbf{s}, \mathbf{w}) - \nabla_{\mathbf{w}} h(\hat{\mathbf{s}}, \hat{\mathbf{w}})\|_2^2 \leq \beta^2 H_w \|\mathbf{w} - \hat{\mathbf{w}}\|_2^2. \end{aligned}$$

*Lemma 3 (Bound dual by primal):* For all  $j \in \mathbb{N}$ ,

$$\begin{aligned} (a) \mathbf{u}_1^j &= -\frac{1}{\rho_1} (\nabla_{\mathbf{s}} h(\mathbf{s}^j, \mathbf{w}^{j-1}) + \nabla_{\mathbf{s}} g(\mathbf{s}^j, \mathbf{t}^{j-1}, \mathbf{w}^{j-1})) \\ &+ \mathbf{p}^{j-1} - \mathbf{p}^j \\ (b) \mathbf{u}_2^j &= -\frac{1}{\rho_2} (\nabla_{\mathbf{w}} h(\mathbf{s}^j, \mathbf{w}^j) + \nabla_{\mathbf{w}} g(\mathbf{s}^j, \mathbf{t}^j, \mathbf{w}^j)) \\ &+ \mathbf{v}^{j-1} - \mathbf{v}^j \\ (c) \|\mathbf{u}_1^j - \mathbf{u}_1^{j+1}\|_2^2 &\leq \frac{1}{\rho_1^2} ((\lambda_1^2 G_s^s + \alpha^2 H_s) \|\mathbf{s}^j - \mathbf{s}^{j+1}\|_2^2 \\ &+ (1/\epsilon^4 + \lambda_1^2 G_s^t) \|\mathbf{t}^{j-1} - \mathbf{t}^j\|_2^2 \\ &+ \rho_1^2 \|\mathbf{p}^{j-1} - \mathbf{p}^j\|_2^2 + \rho_1^2 \|\mathbf{p}^j - \mathbf{p}^{j+1}\|_2^2) \\ (d) \|\mathbf{u}_2^j - \mathbf{u}_2^{j+1}\|_2^2 &\leq \frac{1}{\rho_2^2} ((\lambda_2^2 G_w^w + \beta^2 H_w) \|\mathbf{w}^j - \mathbf{w}^{j+1}\|_2^2 \end{aligned}$$

$$\begin{aligned}
& + \lambda_2^2 G_w^t \|\mathbf{t}^j - \mathbf{t}^{j+1}\|_2^2 \\
& + \rho_2^2 \|\mathbf{v}^{j-1} - \mathbf{v}^j\|_2^2 + \rho_2^2 \|\mathbf{v}^j - \mathbf{v}^{j+1}\|_2^2
\end{aligned}$$

We next list four key properties under which ADMM convergence can be ensured. The Supplementary Material contains proofs that the ADMM steps (15a)–(15g) satisfy the following properties when applied to problem (14).

**Property P1 (Monotone, lower-bounded  $\mathcal{L}_\rho$  and bounded sequence).** The sequence  $\{\mathbf{s}^j, \mathbf{p}^j, \mathbf{t}^j, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j\}$  is bounded, and the Lagrangian  $\mathcal{L}_\rho(\mathbf{s}^j, \mathbf{p}^j, \mathbf{t}^j, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j)$  is lower bounded.

**Property 2 (Sufficient descent).** There is a constant  $C_1(\rho, \lambda) > 0$  such that for all  $k$ , we have that

$$\begin{aligned}
& \mathcal{L}_\rho(\mathbf{s}^0, \mathbf{p}^0, \mathbf{t}^0, \mathbf{w}^0, \mathbf{v}^0, \mathbf{u}_1^0, \mathbf{u}_2^0) \\
& - \mathcal{L}_\rho(\mathbf{s}^j, \mathbf{p}^j, \mathbf{t}^j, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j) \\
& \geq C_1(\rho, \lambda) \sum_{i=0}^j (\|\mathbf{s}^i - \mathbf{s}^{i+1}\|_2^2 + \|\mathbf{p}^i - \mathbf{p}^{i+1}\|_2^2 \\
& + \|\mathbf{t}^i - \mathbf{t}^{i+1}\|_2^2 + \|\mathbf{w}^i - \mathbf{w}^{i+1}\|_2^2 + \|\mathbf{v}^i - \mathbf{v}^{i+1}\|_2^2).
\end{aligned}$$

**Property 3 (Subgradient bound).** There is a constant  $C_2(\rho, \lambda) > 0$  and subdifferential

$$\mathbf{d}^{j+1} \in \partial \mathcal{L}_\rho(\mathbf{s}^{j+1}, \mathbf{p}^{j+1}, \mathbf{t}^{j+1}, \mathbf{w}^{j+1}, \mathbf{v}^{j+1}, \mathbf{u}_1^{j+1}, \mathbf{u}_2^{j+1})$$

such that

$$\begin{aligned}
\|\mathbf{d}^{j+1}\|_2^2 & \leq C_2(\rho, \lambda) (\|\mathbf{s}^j - \mathbf{s}^{j+1}\|_2^2 + \|\mathbf{p}^j - \mathbf{p}^{j+1}\|_2^2 \\
& + \|\mathbf{p}^{j-1} - \mathbf{p}^j\|_2^2 + \|\mathbf{t}^j - \mathbf{t}^{j+1}\|_2^2 + \|\mathbf{t}^{j-1} - \mathbf{t}^j\|_2^2 \\
& + \|\mathbf{w}^j - \mathbf{w}^{j+1}\|_2^2 + \|\mathbf{v}^j - \mathbf{v}^{j+1}\|_2^2 + \|\mathbf{v}^{j-1} - \mathbf{v}^j\|_2^2).
\end{aligned}$$

**Property 4 (Limiting continuity).** If  $(\mathbf{s}^*, \mathbf{p}^*, \mathbf{t}^*, \mathbf{w}^*, \mathbf{v}^*, \mathbf{u}_1^*, \mathbf{u}_2^*)$  is the limit point of the subsequence  $(\mathbf{s}^{j_a}, \mathbf{p}^{j_a}, \mathbf{t}^{j_a}, \mathbf{w}^{j_a}, \mathbf{v}^{j_a}, \mathbf{u}_1^{j_a}, \mathbf{u}_2^{j_a})$  for  $a \in \mathbb{N}$ , then  $\mathcal{L}_\rho(\mathbf{s}^*, \mathbf{p}^*, \mathbf{t}^*, \mathbf{w}^*, \mathbf{v}^*, \mathbf{u}_1^*, \mathbf{u}_2^*)$  is the limit of  $\mathcal{L}_\rho(\mathbf{s}^{j_a}, \mathbf{p}^{j_a}, \mathbf{t}^{j_a}, \mathbf{w}^{j_a}, \mathbf{v}^{j_a}, \mathbf{u}_1^{j_a}, \mathbf{u}_2^{j_a})$  as  $a \rightarrow \infty$ .

Finally, under the above four properties, we can demonstrate the convergence of ADMM applied to problem (14). By Property 1, we have that the sequence  $(\mathbf{s}^j, \mathbf{p}^j, \mathbf{t}^j, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j)$  is bounded and thus has a convergent subsequence  $(\mathbf{s}^{j_a}, \mathbf{p}^{j_a}, \mathbf{t}^{j_a}, \mathbf{w}^{j_a}, \mathbf{v}^{j_a}, \mathbf{u}_1^{j_a}, \mathbf{u}_2^{j_a})$  for  $a \in \mathbb{N}$  with the limit point  $(\mathbf{s}^*, \mathbf{p}^*, \mathbf{t}^*, \mathbf{w}^*, \mathbf{v}^*, \mathbf{u}_1^*, \mathbf{u}_2^*)$  as  $a \rightarrow \infty$ . Under Properties 1 and 2, the Lagrangian  $\mathcal{L}_\rho(\mathbf{s}^j, \mathbf{p}^j, \mathbf{t}^j, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j)$  is bounded above and below, so we have that  $\|\mathbf{s}^j - \mathbf{s}^{j+1}\|_2^2 \rightarrow 0$ ,  $\|\mathbf{p}^j - \mathbf{p}^{j+1}\|_2^2 \rightarrow 0$ ,  $\|\mathbf{t}^j - \mathbf{t}^{j+1}\|_2^2 \rightarrow 0$ ,  $\|\mathbf{w}^j - \mathbf{w}^{j+1}\|_2^2 \rightarrow 0$ ,  $\|\mathbf{v}^j - \mathbf{v}^{j+1}\|_2^2 \rightarrow 0$  as  $j \rightarrow \infty$ . Then, by Lemma 3 and the primal variable convergence, we also have that  $\|\mathbf{u}_1^j - \mathbf{u}_1^{j+1}\|_2^2 \rightarrow 0$  and  $\|\mathbf{u}_2^j - \mathbf{u}_2^{j+1}\|_2^2 \rightarrow 0$  as  $j \rightarrow \infty$ . Based on Property 3, there exists a subgradient sequence  $\mathbf{d}^j \in \partial \mathcal{L}_\rho(\mathbf{s}^j, \mathbf{p}^j, \mathbf{t}^j, \mathbf{w}^j, \mathbf{v}^j, \mathbf{u}_1^j, \mathbf{u}_2^j)$  such that  $\|\mathbf{d}^j\|_2^2 \rightarrow 0$  as  $j \rightarrow \infty$ . Thus, we also have that  $\|\mathbf{d}^{j_a}\|_2^2 \rightarrow 0$  as  $a \rightarrow \infty$ . Finally, by Property 4, we have that  $\mathcal{L}_\rho(\mathbf{s}^*, \mathbf{p}^*, \mathbf{t}^*, \mathbf{w}^*, \mathbf{v}^*, \mathbf{u}_1^*, \mathbf{u}_2^*)$  is the limit of  $\mathcal{L}_\rho(\mathbf{s}^{j_a}, \mathbf{p}^{j_a}, \mathbf{t}^{j_a}, \mathbf{w}^{j_a}, \mathbf{v}^{j_a}, \mathbf{u}_1^{j_a}, \mathbf{u}_2^{j_a})$  as  $a \rightarrow \infty$ . By the definition of the general subgradient, we have that  $0 \in \partial \mathcal{L}_\rho(\mathbf{s}^*, \mathbf{p}^*, \mathbf{t}^*, \mathbf{w}^*, \mathbf{v}^*, \mathbf{u}_1^*, \mathbf{u}_2^*)$ .

## REFERENCES

- [1] M. Navarro and S. Segarra, "Graphon-aided joint estimation of multiple graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5458–5462.
- [2] I. Brugere, B. Gallagher, and T. Y. Berger-Wolf, "Network structure inference, a survey: Motivations, methods, and applications," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 1–39, 2018.
- [3] D. R. Farine and H. Whitehead, "Constructing, conducting and interpreting animal social network analysis," *J. Animal Ecol.*, vol. 84, no. 5, pp. 1144–1163, 2015.
- [4] M. Narayan and G. I. Allen, "Mixed effects models for resampled network statistics improves statistical power to find differences in multi-subject functional connectivity," *Front. Neurosci.*, vol. 10, 2016, Art. no. 108.
- [5] J. F. Donges, Y. Zou, N. Marwan, and J. Kurths, "Complex networks in climate dynamics," *Eur. Phys. J. Special Topics*, vol. 174, no. 1, pp. 157–179, 2009.
- [6] P. Das and A. Ortega, "Graph-based skeleton data compression," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, 2020, pp. 1–6.
- [7] O. Sporns, "The human connectome: Origins and challenges," *NeuroImage*, vol. 80, pp. 53–61, 2013.
- [8] R. M. Hutchison et al., "Dynamic functional connectivity: Promise, issues, and interpretations," *NeuroImage*, vol. 80, pp. 360–378, 2013.
- [9] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. New York, NY, USA: Springer, 2009.
- [10] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, May 2019.
- [11] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [12] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the lasso," *Ann. Statist.*, vol. 34, pp. 1436–1462, 2006.
- [13] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLoS Comput. Biol.*, vol. 9, no. 5, 2013, Art. no. e1003068.
- [14] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 920–929.
- [15] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.
- [16] T. M. Roddenberry, M. Navarro, and S. Segarra, "Network topology inference with graphon spectral penalties," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5390–5394.
- [17] S. de Silva, A. D. Ranjeewa, and S. Kryazhimskiy, "The dynamics of social networks among female Asian elephants," *BMC Ecol.*, vol. 11, no. 1, pp. 1–16, 2011.
- [18] L. Lovász, *Large Networks and Graph Limits*, vol. 60. Providence, RI, USA: Amer. Math. Soc., 2012.
- [19] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.
- [20] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.
- [21] 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, Nov. 2017.
- [22] P. Diaconis and S. Janson, "Graph limits and exchangeable random graphs," 2007, *arXiv:0712.2749*.
- [23] M. Avella-Medina, F. Parise, M. T. Schaub, and S. Segarra, "Centrality measures for graphons: Accounting for uncertainty in networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 520–537, Jan./Mar. 2020.
- [24] P. Erdős and A. Rényi, "On random graphs I," *Publicationes Mathematicae*, vol. 6, no. 3–4, pp. 290–297, Nov. 1959.
- [25] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Netw.*, vol. 5, no. 2, pp. 109–137, 1983.
- [26] Y. Wang, S. Segarra, and C. Uhler, "High-dimensional joint estimation of multiple directed gaussian graphical models," *Electron. J. Statist.*, vol. 14, no. 1, pp. 2439–2483, 2020.
- [27] C. Gao, Y. Zhu, X. Shen, and W. Pan, "Estimation of multiple networks in gaussian mixture models," *Electron. J. Statist.*, vol. 10, 2016, Art. no. 1133.
- [28] J. Guo, E. Levina, G. Michailidis, and J. Zhu, "Joint estimation of multiple graphical models," *Biometrika*, vol. 98, no. 1, pp. 1–15, 2011.

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

[30] C. F. W. Peeters, A. E. Bilgrau, and W. N. van Wieringen, "rags2ridges: A one-stop- $\ell_2$ -shop for graphical modeling of high-dimensional precision matrices," *J. Stat. Soft.*, vol. 102, no. 4, pp. 1–32, May 2022.

[31] S. Rey, A. Buculea, M. Navarro, S. Segarra, and A. G. Marques, "Joint inference of multiple graphs with hidden variables from stationary graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5817–5821.

[32] H. Araghi, M. Sabbaqi, and M. Babaie-Zadeh, " $K$ -graphs: An algorithm for graph signal clustering and multiple graph learning," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1486–1490, Oct. 2019.

[33] A. Karaaslanli, S. Saha, S. Aviyente, and T. Maiti, "Multiview graph learning for single-cell RNA sequencing data," *bioRxiv*, 2021, Art. no. 467476, doi: [10.1101/2021.11.05.467476](https://doi.org/10.1101/2021.11.05.467476).

[34] M. Navarro, Y. Wang, A. G. Marques, C. Uhler, and S. Segarra, "Joint inference of multiple graphs from matrix polynomials," *J. Mach. Learn. Res.*, vol. 23, pp. 1–35, 2020.

[35] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, "Network inference via the time-varying graphical lasso," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 205–213.

[36] A. J. Gibberd and J. D. B. Nelson, "Regularized estimation of piecewise constant gaussian graphical models: The group-fused graphical lasso," *J. Comput. Graph. Statist.*, vol. 26, no. 3, pp. 623–634, 2017.

[37] S. Zhou, J. Lafferty, and L. Wasserman, "Time varying undirected graphs," *Mach. Learn.*, vol. 80, no. 2–3, pp. 295–319, 2010.

[38] H. Yu, S. Wu, and J. Dauwels, "Efficient variational bayes learning of graphical models with smooth structural changes," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 2022, doi: [10.1109/TPAMI.2022.3140886](https://doi.org/10.1109/TPAMI.2022.3140886).

[39] H. Araghi, M. Babaie-Zadeh, and S. Achard, "Dynamic  $K$ -graphs: An algorithm for dynamic graph learning and temporal graph signal clustering," in *Proc. IEEE Eur. Signal Process. Conf.*, 2021, pp. 2195–2199.

[40] K. Yamada and Y. Tanaka, "Temporal multiresolution graph learning," *IEEE Access*, vol. 9, pp. 143734–143745, 2021.

[41] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2826–2830.

[42] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning with constraints on graph temporal variation," 2020, *arXiv:2001.03346*.

[43] J. Hong and X. Dai, "Demixing and topology identification for mixed graph signals," in *Proc. IEEE 5th Int. Conf. Commun. Inf. Syst.*, 2021, pp. 154–159.

[44] I. Ricchi, A. Tarun, H. P. Maretic, P. Frossard, and D. Van De Ville, "Dynamics of functional network organization through graph mixture learning," *NeuroImage*, vol. 252, 2022, Art. no. 119037.

[45] H. P. Maretic and P. Frossard, "Graph Laplacian mixture model," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 6, pp. 261–270, 2020.

[46] J.-Y. Kao, D. Tian, H. Mansour, A. Ortega, and A. Vetro, "Disc-GLasso: Discriminative graph learning with sparsity regularization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2956–2960.

[47] B. Hao, W. W. Sun, Y. Liu, and G. Cheng, "Simultaneous clustering and estimation of heterogeneous graphical models," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 7981–8038, 2017.

[48] A. Lotsi and E. Wit, "Sparse Gaussian graphical mixture model," *Afrika Statistika*, vol. 11, no. 2, pp. 1041–1059, 2016.

[49] K. Tsai, O. Koyejo, and M. Kolar, "Joint Gaussian graphical model estimation: A survey," *WIREs Comput. Statist.*, vol. 14, no. 6, 2022, Art. no. e1582.

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

[51] E. Pircalabelu, G. Claeskens, and L. J. Waldorp, "Mixed scale joint graphical lasso," *Biostatistics*, vol. 17, no. 4, pp. 793–806, 2016.

[52] J. Arroyo, A. Athreya, J. Cape, G. Chen, C. E. Priebe, and J. T. Vogelstein, "Inference for multiple heterogeneous networks with a common invariant subspace," *J. Mach. Learn. Res.*, vol. 22, no. 142, pp. 1–49, 2021.

[53] I. Soloveychik and A. Wiesel, "Joint estimation of inverse covariance matrices lying in an unknown subspace," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2379–2388, May 2017.

[54] Q. Tao, X. Huang, S. Wang, X. Xi, and L. Li, "Multiple Gaussian graphical estimation with jointly sparse penalty," *Signal Process.*, vol. 128, pp. 88–97, 2016.

[55] K. Mohan, P. London, M. Fazel, D. Witten, and S.-I. Lee, "Node-based learning of multiple Gaussian graphical models," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 445–488, 2014.

[56] C. Gao, Y. Lu, and H. H. Zhou, "Rate-optimal graphon estimation," *Ann. Statist.*, vol. 43, no. 6, pp. 2624–2652, 2015.

[57] O. Klopp and N. Verzelen, "Optimal graphon estimation in cut distance," *Probability Theory Related Fields*, vol. 174, no. 3, pp. 1033–1090, 2019.

[58] B. Sischka and G. Kauermann, "EM-based smooth graphon estimation using MCMC and spline-based approaches," *Social Netw.*, vol. 68, pp. 279–295, 2022.

[59] J. Yang, C. Han, and E. Airoldi, "Nonparametric estimation and testing of exchangeable graph models," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2014, pp. 1060–1067.

[60] P. Latouche and S. Robin, "Variational bayes model averaging for graphon functions and motif frequencies inference in W-graph models," *Statist. Comput.*, vol. 26, no. 6, pp. 1173–1185, 2016.

[61] S. Chan and E. Airoldi, "A consistent histogram estimator for exchangeable graph models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 208–216.

[62] H. Xu, D. Luo, L. Carin, and H. Zha, "Learning graphons via structured gromov-wasserstein barycenters," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10505–10513.

[63] P. J. Wolfe and S. C. Olhede, "Nonparametric graphon estimation," 2013, *arXiv:1309.5936*.

[64] S. C. Olhede and P. J. Wolfe, "Network histograms and universality of blockmodel approximation," *Proc. Nat. Acad. Sci.*, vol. 111, no. 41, pp. 14722–14727, 2014.

[65] E. M. Airoldi, T. B. Costa, and S. H. Chan, "Stochastic blockmodel approximation of a graphon: Theory and consistent estimation," *Adv. Neural Inf. Process. Syst.*, vol. 26, pp. 692–700, 2013.

[66] D. S. Choi, P. J. Wolfe, and E. M. Airoldi, "Stochastic blockmodels with a growing number of classes," *Biometrika*, vol. 99, no. 2, pp. 273–284, 2012.

[67] K. Rohe, S. Chatterjee, and B. Yu, "Spectral clustering and the high-dimensional stochastic blockmodel," *Ann. Statist.*, vol. 39, no. 4, pp. 1878–1915, 2011.

[68] Z. Zhao, L. Chen, and L. Lin, "Change-point detection in dynamic networks via graphon estimation," 2019, *arXiv:1908.01823*.

[69] Y. Zhang, E. Levina, and J. Zhu, "Estimating network edge probabilities by neighbourhood smoothing," *Biometrika*, vol. 104, no. 4, pp. 771–783, 2017.

[70] Y. Qin, L. Yu, and Y. Li, "Iterative connecting probability estimation for networks," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 1155–1166, 2021.

[71] D. Song, C. E. Lee, Y. Li, and D. Shah, "Blind regression: Nonparametric regression for latent variable models via collaborative filtering," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 2155–2163, 2016.

[72] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[73] C. Borgs, J. Chayes, C. E. Lee, and D. Shah, "Thy friend is my friend: Iterative collaborative filtering for sparse matrix estimation," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 4715–4726, 2017.

[74] S. Chatterjee, "Matrix estimation by universal singular value thresholding," *Ann. Statist.*, vol. 43, no. 1, pp. 177–214, 2015.

[75] J. Duchon, "Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces," *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, vol. 10, no. R3, pp. 5–12, 1976.

[76] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proc. ACM Nat. Conf.*, 1968, pp. 517–524.

[77] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. Delft, The Netherlands: Now Publishers Inc, 2011.

[78] F. Klimm, N. S. Jones, and M. T. Schaub, "Modularity maximisation for graphons," 2021, *arXiv:2101.00503*.

[79] S. P. MacLachlan and L. N. Olson, "Theoretical bounds for algebraic multigrid performance: Review and analysis," *Numer. Linear Algebra Appl.*, vol. 21, no. 2, pp. 194–220, 2014.

[80] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

- [81] H. A. L. Thi and T. P. Dinh, "The DC (difference of convex functions) programming and DCA revisited with DC models of real world non-convex optimization problems," *Ann. Operations Res.*, vol. 133, no. 1–4, pp. 23–46, 2005.
- [82] R. Horst and N. V. Thoai, "DC programming: Overview," *J. Optim. Theory Appl.*, vol. 103, no. 1, pp. 1–43, 1999.
- [83] S. Chandra, S. Olhede, and P. Wolfe, "Local linear graphon estimation using covariates," *Biometrika*, vol. 109, no. 3, pp. 721–734, Sep. 2022.
- [84] J. B. Lewis, K. Poole, H. Rosenthal, A. Boche, A. Rudkin, and L. Sonnet, "Voteview: Congressional roll-call votes database," 2022. [Online]. Available: <https://voteview.com/>
- [85] E. S. Lein et al., "Genome-wide atlas of gene expression in the adult mouse brain," *Nature*, vol. 445, no. 7124, pp. 168–176, 2007.
- [86] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, 2019.



**Madeline Navarro** (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from Old Dominion University, Norfolk, VA, USA, in 2019. She is currently working toward the Ph.D. degree in electrical and computer engineering with Rice University, Houston, TX, USA. Her research interests include graph signal processing, graphical models, and graph-based machine learning.



**Santiago Segarra** (Senior Member, IEEE) received the B.Sc. degree (with highest hon.) in industrial engineering (Valedictorian) from the Instituto Tecnológico de Buenos Aires, Buenos Aires, Argentina, in 2011, the M.Sc. degree in electrical engineering from the University of Pennsylvania (Penn), Philadelphia, PA, USA, in 2014, and the Ph.D. degree in electrical and systems engineering from Penn in 2016. From September 2016 to June 2018, he was a Postdoctoral Research Associate with the Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA, USA. Since July 2018, he has been a W. M. Rice Trustee Assistant Professor with the Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA. His research interests include network theory, data analysis, machine learning, and graph signal processing. Dr. Segarra was the recipient of the ITBA's 2011 Best Undergraduate Thesis Award in Industrial Engineering, 2011 Outstanding Graduate Award granted by the National Academy of Engineering of Argentina, 2017 Penn's Joseph and Rosaline Wolf Award for Best Doctoral Dissertation in electrical and systems engineering, 2020 IEEE Signal Processing Society Young Author Best Paper Award, 2021 Rice's School of Engineering Research + Teaching Excellence Award, and five best conference paper awards.