Abstract

The first step for any graph signal processing (GSP) procedure is to learn the graph signal representation, i.e., to capture the dependence structure of the data into an adjacency matrix. Indeed, the adjacency matrix is typically not known a priori and has to be learned. However, it is learned with errors. A little attention has been paid to modelling such errors in the adjacency matrix, and studying their effects on GSP methods. However, modelling errors in the adjacency matrix will enable both to study the graph error effects in GSP and to develop robust GSP algorithms. In this paper, we therefore introduce practically justifiable graph error models. We also study, both analytically when possible and numerically, the graph error effect on the performance of GSP methods in different types of problems such as filtering of graph signals and independent component analysis of graph signals (graph decorrelation).

Keywords: Erdös-Rényi graphs, error effect, graph signal processing, minimum distance index, shift matrix

I. INTRODUCTION

In the classical signal processing setup where the digital signals are represented in terms of time series or vectors of spatial measurements (for example, measurements by sensor arrays), it is assumed that each point of a discrete signal depends on the preceding or spatially close point of the signal. With the current advances in the data collection and representation, when
the data points may not be ordered temporally or spatially and the digital signal may no longer be accurately or even adequately represented by a simple time series structure, the classical signal processing tools are no longer applicable. Thus, graph signal processing (GSP) [1]–[3] has emerged as a new area of signal processing and data analysis where one of the major aims is to generalize the standard signal processing methods and concepts into the context of more complex signals represented on graphs that can have various structures. The graph corresponding to ordered data is then just a simple special case of a digital signal graph. Examples of digital signals represented on graphs include sensor networks, brain networks, gene regulatory networks, and social networks to name just a few most frequently encountered graph signals [4]–[7].

In GSP, each signal value is indexed by a node in a graph at which this signal value is measured, and edges between pairs of nodes in a graph indicate dependencies between the corresponding signal values. Then the underlying graph can be fully specified by an adjacency matrix, denoted hereafter as $A$, whose $(i, j)$th element is nonzero if the $i$th and the $j$th nodes are connected, and the value $[A]_{i,j} = a_{ij}$ describes the strength of the relationship. The adjacency matrix can be viewed as the basis of GSP, on which for example the graph Fourier transform (GFT) and graph filters [1] are built, often via the graph Laplacian matrix or other shift operators which are based on the adjacency matrix [1].

The process of obtaining the adjacency matrix may take different forms. There may exist physical or friendship connections between the nodes known in advance, which directly yield a graph. At the other extreme end, there may be no auxiliary information available about the connectedness of the nodes, and the methods for finding an adjacency matrix are then entirely based on training data. Methods for estimating the graph Laplacian matrix in the latter case have been developed in [14]–[18]. Third, the choice of the adjacency matrix may be based on other variables, which are not perfectly correlated with the graph signal which we are interested in. In both the second and the third cases, the process of adjacency matrix learning is necessarily stochastic. As a result, there are multiple sources of errors which lead to imperfect learning of the adjacency matrix. Even in the case when the choice of the adjacency matrix seems to be obvious and based on the known physical or friendship connections between the nodes, such

---

1The GSP literature is developing fast and by now covers also sampling theory for graph signals [8], [9], stationarity theory for graph signals [10], and percolation [11] of graph signals. Graph filters have been applied for example in distributed average consensus problem [12], [13].
choice still may not be the best one.

Our focus in this paper is to study consequences of imperfect specifications of graph signal adjacency matrix to GSP tasks. The source of errors in the adjacency matrix learning may be related to or be completely irrelevant to the sources of errors in the signal measurements. However, even when the sources of errors in the adjacency matrix learning and signal value measurements are the same, the effect of errors in the adjacency matrix on GSP tasks are different from those caused by the errors in signal measurements. Considerations of the graph learning errors has only recently gained some attention. For example, a discussion on the topic is given in [19], where eigen-decomposition of the graph Laplacian was analyzed when small subsets of edges are added or deleted. Based on the analysis, robust methods were built for GSP tasks such as signal recovery, label propagation and clustering, when the knowledge of edgewise probabilities of errors in the graph topology is available.

There are also some works which study robustness with respect to graph topology errors in specific tasks. For example, total least-squares approach for robust graph signal recovery was proposed in [20]. Graph neural networks were found to be stable to small changes in graph topology in [21]. Node-variant graph filters were introduced in [22], and they were shown by numerical experiments to be more robust to errors in graph shift operator than node-invariant filters. For distributed average consensus problem, [23] proposed a method which optimizes worst case performance when the eigenvalue distribution is considered uncertain. In [24], a new centrality measure was introduced and the robustness with respect to edge weight perturbation was considered. The robustness to some missing edges was also part of the comparison of autoregressive moving average graph filters and finite impulse response graph filters for time-varying graph signals in [25]. For spectral graph filters, [26] defined Caley smoothness filter space where the filter perturbation is bounded by a constant times the perturbation of the graph. In the presence of erroneous edges, [27] studied differences between small world graph and Erdős–Rényi graph, and [28] studied clustering of nodes into two communities.

The starting point for studying GSP performance under the condition of imperfect knowledge of graph signal adjacency matrix is the modelling of graph errors. As the errors in the signal modelling in the classical signal processing may lead to significant performance degradation (see for example [29]), the errors in the adjacency matrix may have a significant effect on the GSP performance. Then the development of robust GSP methods, just as the development of
robust signal processing methods [29], is of a great importance. This paper, to the best of our knowledge, is the first attempt towards developing justifiable and generic enough models for adjacency matrix mismatches. It also studies the effects of the mismatched adjacency matrix on the performance of some more traditional GSP applications such as filtering of graph signals as well as independent component analysis (ICA) of graph signals (also referred to as graph decorrelation (GraDe) in the context of separating signals based on graph structure only).

A. Contributions

Our main contributions are the following:

- We formulate graph error models for the adjacency matrix, which help to quantify the deviation from the true matrix using a few parameters. We consider adding and removing edges and perturbing the edge weights. First, we build models assuming equal probabilities of mislearning the edge status for each pair of nodes, but later we generalize to the case where there are several subsets of pairs such that within the groups the probabilities are equal, but they differ between the groups.
- We study the structural effects of the proposed error models on the adjacency matrix, including spectrum analysis where applicable and possible.
- We introduce the graph autocorrelation, which is a scale and edge density invariant measure of smoothness between the signal and the graph.
- We illustrate what effects different type of errors in adjacency matrix specification produce in filtering of graph signal and ICA of graph signals. Some theoretical studies as well as numerical and real data-based studies are performed.

B. Notation

We use boldface capital letters for matrices, boldface lowercase letters for vectors, and capital calligraphic letters for sets. The exceptions are 1_N which is the N-dimensional vector full of ones, the M × N matrix full of ones 1_M×N = 1_M1_N^T, and 1_A is a matrix of the same size as A, such that [1_A]_{i,j} = 1, if a_{i,j} ≠ 0 and [1_A]_{i,j} = 0, if a_{i,j} = 0. The matrix I_{N×N} is the N × N identity matrix. The notations (·)^T, ⊙, ∥·∥, tr{·}, P(·), E{·}, and var(·) stand for

2Our preliminary work on the topic has been reported in [30].
the transpose, Hadamard product, Euclidian norm of a vector, trace of a matrix, probability, mathematical expectation, and variance, respectively. The notation for diagonal elements of a matrix $A$ is $\text{diag}(A)$, and the notation $N(0, \sigma^2)$ stands for Gaussian zero-mean distribution with variance $\sigma^2$.

C. Paper Organization

In Section II, we start by recalling graph models and establishing a result, which brings an analogy between Gaussian distribution and Erdős–Rényi model. This motivates the special role of Erdős–Rényi model in graph error models which are introduced for different types of graphs. Section III is devoted to defining and studying graph autocorrelation. In Section IV, the effect of adjacency matrix mismatch for several GSP applications is studied using simulations and real data examples. Section V concludes the paper.

II. GRAPH ERROR MODELS

A. Basic Building Blocks

Let us first go through the basics of graphs, and define the graph models which will be used in the paper.

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be a directed graph that represents the basis of a graph signal, where $\mathcal{N}$ is the set of $N$ nodes and $\mathcal{E}$ is the set of edges. The adjacency matrix of the graph $\mathcal{G}$, denoted as $A$, is a matrix that satisfies the conditions $a_{i,i} = 0$ for $i = 1, \ldots, N$ and $a_{i,j} \neq 0$ if and only if $(i, j) \in \mathcal{E}$, i.e., there is an edge from node $j$ to node $i$.

For developing our graph error models, we will use the Erdős–Rényi model according to which a random graph is constructed by connecting nodes randomly with a constant probability $[31]$. The corresponding graph is denoted as $\mathcal{G} = (\mathcal{N}, \epsilon)$ and its adjacency matrix $\Delta_\epsilon$ is a random $N \times N$ matrix such that $\mathbb{P}([\Delta_\epsilon]_{i,j} = 1) = \epsilon$ and $\mathbb{P}([\Delta_\epsilon]_{i,j} = 0) = 1 - \epsilon$ for all $i \neq j$, and $[\Delta_\epsilon]_{i,i} = 0$ for $i = 1, \ldots, N$, where each element of the matrix is generated independently from the other elements.

Another graph model which will be used in this paper is the stochastic block model (SBM) $[32]$, where each node belongs to one of $r$ communities and connections within a community are more common than connections between nodes in different communities. Let $N_1, \ldots, N_r$ denote the sizes of the communities and let us assume that the nodes are ordered so that the $i$th node...
belongs to the \( j \)th community if and only if \( \sum_{i=1}^{j-1} N_l < i \leq \sum_{l=1}^{j} N_l \). The probability of the edge existence from a node of the \( j \)th community to a node of the \( i \)th community is denoted as \( p_{i,j} \). A special case of the stochastic block model when \( p_{i,i} = p_{j,j} = p \) for all \( i \) and \( j \), and \( p_{i,j} = p_{k,l} = q \) for any \( i \neq j \) and \( k \neq l \), is called planted partition model (PPM).

Because of the errors in the adjacency matrix learning, the estimated adjacency matrix deviates from the true one. In the following subsections, we introduce graph error models in the form aligned with the objective to study the effect of the errors in the adjacency matrix. All models share the same additive structure, i.e., have the form

\[
W = A + E, \tag{1}
\]

where \( W \) presents the estimated adjacency matrix, \( A \) is the correct adjacency matrix, and \( E \) is an unknown error matrix which can be viewed as an analog of the additive error/distortion/noise component (that applies not to the signal points, but to the signal structure) in the traditional signal processing and time series analysis. Notice that \( E \) needs to be a function of \( A \) because it depends on \( A \) whether distorting an edge means adding or deleting it. Moreover, the Erdős–Rényi graph is the basic GSP error/distortion/noise model analogous to the basic Gaussian noise model in the traditional signal processing as will be shown in the next subsection.

The correct adjacency matrix for a given graph signal and a GSP method can be defined as the one which yields the best results. In the case when the data generating process is known, the optimal adjacency matrix in that sense can often be found. On the other hand, when the optimal adjacency matrix is not known, we might still want to use the graph error models to see how much the results vary when the adjacency matrix is perturbed. Then the role of correct adjacency matrix \( A \) can be assigned to an estimated adjacency matrix as well.

### B. Motivation for Erdős-Rényi Graph Error Models

In the context of our paper, an important characteristic of the Erdős-Rényi graph is that it does not allow for formation of communities [33], and if applied on the top of another graph, it will not change the essential structure of that graph. Instead, it just disturbs the spectrum of the original graph as will be shown later in this section.

The simplicity and the above-mentioned properties are the main reasons for the use of Erdős-Rényi model as the main block for modelling the mismatches in the graphical structures of graph
signals. However, we will further justify it by bringing an analogy to the basic nature of the Gaussian additive noise for modelling errors related to imperfect fit between signal models and actual measured signal values.

Similar to the fact that the key for modelling additive noise is the noise probability density function, the key for modelling graph errors will be the graph function or graphon. The term graphon was introduced in [34], in the context of graph sequence theory, and it refers to measurable and symmetric functions mapping from \([0, 1]^2\) to \([0, 1]\). Graphons have been used, for example, as graph limit objects [34], [35], and as building blocks in the kernel graph model [36]. Moreover, they have been used for spectral analysis of the adjacency matrix [33]. See Appendix A for a brief introduction to graphons.

In traditional signal processing, the main reason for the use of Gaussian random variables as error terms meant to model the discrepancies between the analytic signal model and actual measurements is the central limit theorem (CLT). This theorem states that the distribution of sum of independent and identically distributed random variables tends to the Gaussian distribution as the number of summands grows. With respect to modelling errors in graphs, the Erdős-Rényi graph plays a special role due to its simplicity. However, the importance of the Erdős-Rényi graph also follows from the fact that similar to the Gaussian distribution for measurement error modelling, the Erdős-Rényi graphon leads to a theorem that can be considered as an analog of CLT in the case of modelling the graph structure errors.

Let \(W_\epsilon\) denote the set of all graphons \(W\), which is a weighted graph with uncountable set of nodes, satisfying

\[
\int_0^1 \int_0^1 W(x, y) dx dy = \epsilon
\]

where \(W(x, y)\) is the edge weight between nodes \(x\) and \(y\) in the graphon \(W\) (see Appendix A for more details), and \(0 < \epsilon < 1\). Then the following theorem is in order.

**Theorem 1.** Let \(W_1, \ldots, W_M\) be random samples from \(W_\epsilon\) and \(c_1, \ldots, c_M\) be random samples from a distribution with support on \([0, 1]\) and expected value \(c\). Assume that \(W_1, \ldots, W_M, c_1, \ldots, c_M\) are mutually independent. Then for all \(0 \leq x, y \leq 1\), we almost surely have

\[
\bar{W}(x, y) = \frac{1}{M} \sum_{i=1}^M c_i W_i(x, y) \to c \epsilon \quad \text{as} \quad M \to \infty.
\]

The proof of the theorem is given at the end of Appendix A.
Theorem 1 states that the average of many independent graphons converges to a constant, i.e., to the Erdős-Rényi graphon. Thus, the theorem further motivates the use of Erdős-Rényi graphs as the main building block in developing graph error models for GSP, when the summands are considered as noise factors which impede the estimation of the graph structure. In this sense, Theorem 1 is an analog of the CLT.

C. A Basic Model for Unweighted Graphs

We start by considering unweighted graphs, for which the adjacency matrix becomes \( A = 1_A \). First, making a somewhat idealistic assumption that the outcome of the graph signal adjacency matrix learning is accurate enough, that is, assuming that incorrect graph edge learning is equally probable for any edge in the graph, the learning errors can be simply modelled by the Erdős-Rényi graph. Then the actually available learned adjacency matrix of a graph signal can be modelled in terms of the following inaccurate version of \( A \)

\[
W = A + \Delta_\epsilon \odot (1_{N \times N} - 2A).
\]  

(M1)

According to (M1), the true adjacency matrix of a graph signal is distorted because of the imperfect learning, and this distortion follows the Erdős-Rényi model, where the level of distortion depends on a single parameter, which is the probability \( \epsilon \). As a result of the distortion captured by model (M1), an edge can be added with probability \( \epsilon \), when there is no edge in the true graph, or an edge of the true graph can be removed with the same probability, if there exists an edge in the true graph. It corresponds to flipping the value from 0 to 1 or from 1 to 0 in the adjacency matrix \( 1_A \) in the positions corresponding to value 1 in the Erdős-Rényi adjacency matrix \( \Delta_\epsilon \). This basic model is introduced due to its simplicity as it has a single parameter, \( \epsilon \). However, it may not be sufficiently flexible to describe signal graph errors that appear as outcome in any practical graph learning process. For example, it will change the number of edges significantly if \( \epsilon \) is large and the number of edges is small.

Model (M1) can be easily modified/revised for applying to undirected graphs by defining lower triangular matrices \( \Delta^l_\epsilon \) analogously to \( \Delta_\epsilon \), and then replacing \( \Delta_\epsilon \) by \( \Delta^l_\epsilon + (\Delta^l_\epsilon)^\top \). It is also worth noting that all the following graph error models will be given for directed graphs.

Even though the results in [31] are for undirected graphs only and often Erdős-Rényi model is considered to imply undirected graph, we will use the term Erdős-Rényi for both undirected and directed graphs.
and the same technique as above can be used to derive the corresponding model for undirected graphs.

**D. Different Probabilities of Missed and Mislearned Edges**

Basic model (M1) can be easily extended to the case where the probability of removing an edge as a result of distortion from a graph which correctly captures a graph signal, denoted as $\epsilon_1$, is not the same as the probability of adding an edge, which does not exist in the true graph, denoted as $\epsilon_2$. The corresponding inaccurately learned adjacency matrix of a graph signal can then be modelled as

$$W = A - \Delta_{\epsilon_1} \odot A + \Delta_{\epsilon_2} \odot (1_{N \times N} - A).$$

(M2)

The second term on the right-hand side in (M2) corresponds to edge removal and the third term corresponds to edge addition. This is a straightforward extension of basic model (M1), which is, however, very useful for modelling the typical situation in the graph structure learning when particular algorithms are more or less likely to miss an actually existing edge rather than to mislearn an actually non-existing edge [14]. Model (M2) can be interpreted as an application of two Erdős–Rényi graphs on the top of the true graph, where one Erdős–Rényi graph $\mathcal{G} = (\mathcal{N}, \epsilon_2)$ can only add edges which do not exist in the true graph, while the other Erdős-Rényi graph $\mathcal{G} = (\mathcal{N}, \epsilon_1)$ can only erroneously remove actually existing edges. It is easy to see that model (M2) is equivalent to model (M1) when $\epsilon_1 = \epsilon_2 = \epsilon$.

To analyze the perturbation that the error above proposed models produce, let us start with a trivial observation that if $A$ is given by an Erdős-Rényi graph with probability parameter $\alpha$, then $W$ from model (M2) is also an Erdős-Rényi graph, and the parameter takes value $\alpha(1 - \epsilon_1) + (1 - \alpha)\epsilon_2$.

Moreover, if $A$ is given by the SBM, after applying graph error model (M2), the SBM and the communities in the SBM remain the same as is shown in the following proposition.

**Proposition 1.** Assume that $A$ follows the SBM with probability $p^A_{i,j}$ for edges from a node in the $j$th community to a node in the $i$th community. Then $W$ from graph error model (M2) follows SBM with the same communities as in $A$ and the probabilities are given by

$$p^W_{i,j} = p^A_{i,j}(1 - \epsilon_1) + (1 - p^A_{i,j})\epsilon_2 = (1 - \epsilon_1 - \epsilon_2)p^A_{i,j} + \epsilon_2.$$
Proof: For $W$ given by (M2), the probability $p^W_{i,j}$ for each edge from the $j$th community to the $i$th community is given by the probability that “the edge is in $A$ ($p^A_{i,j}$) and not removed”, i.e., the probability $(1 - \epsilon_1)$, plus the probability that “the edge is not in $A$ ($1 - p^A_{i,j}$) and is added,” i.e., the probability $(\epsilon_2)$. Since $\Delta_{\epsilon_1}$ and $\Delta_{\epsilon_2}$ in (M2) are independent from $A$, the two probabilities above are products of their component probabilities. This observation gives the stated result and completes the proof. □

The spectrum of the adjacency matrix can be found in closed form for simple models such as the undirected (for the error model of undirected graphs, see the end of the next subsection) PPM with equal community sizes [37]. Thus, we also have the following proposition for the spectrum perturbation because of the graph error given by (M2).

**Proposition 2.** Assume that $A$ follows the PPM with $M$ equally large communities, and let $p$ be the probability for edges within communities and $q$ be the probability for edges between communities. If $W$ follows graph error model (M2), the expected difference between the largest eigenvalues of the true adjacency matrix $A$ and the mismatched one $W$ is given by

$$\mathbb{E}\{\lambda_1(A) - \lambda_1(W)\} = \frac{N}{M}(\epsilon_1 + \epsilon_2)(p + (M - 1)q) - M\epsilon_2.$$ 

Moreover, the expected difference between the second to $M$th largest eigenvalues for $A$ and $W$, are mutually equal and given by

$$\mathbb{E}\{\lambda_k(A) - \lambda_k(W)\} = \frac{N}{M}(\epsilon_1 + \epsilon_2)(p - q), \quad k = 2, \ldots, M.$$ 

Proof: For true adjacency matrix $A$, the expected value of the largest eigenvalue is $N(p + (M - 1)q)/M$ and the expected values of next $M - 1$ largest eigenvalues are $N(p - q)/M$ [37]. From Proposition [1] we see that the expected value of the largest eigenvalue of $W$ becomes

$$\mathbb{E}\{\lambda_1(W)\} = \frac{N}{M}((1 - \epsilon_1 - \epsilon_2)p + \epsilon_2 + (M - 1)((1 - \epsilon_1 - \epsilon_2)q + \epsilon_2))$$

and the the expected values of next $M - 1$ largest eigenvalues become

$$\mathbb{E}\{\lambda_2(W)\} = \cdots = \mathbb{E}\{\lambda_{M-1}(W)\} = \mathbb{E}\{\lambda_M(W)\}$$

$$= \frac{N}{M}((1 - \epsilon_1 - \epsilon_2)p + \epsilon_2 - (1 - \epsilon_1 - \epsilon_2)q - \epsilon_2)$$

$$= N((1 - \epsilon_1 - \epsilon_2)(p - q)).$$
Combining these two results together, the difference of the corresponding expected values can be found to be as it is stated in the proposition. It completes the proof. □

The relative change of the largest eigenvalue is thus $\epsilon_1 + (1 - M/(p + (M - 1)q))\epsilon_2$ and the relative change of next $M - 1$ largest eigenvalues is $\epsilon_1 + \epsilon_2$.

E. Generalized Graph Error Model

It is assumed in models (M1) and (M2) that mislearning the edge status is equally probable (although the probabilities of adding non-existing edge and removing existing edge may be different) for all pairs of nodes. For some adjacency matrix estimation methods, this assumption might not hold strictly even if the pairwise connections were equally strong. To allow the differences in learning accuracy, caused by the structure of the graph or the connectivity strength differences, we formulate the following generalized graph error model.

Consider an unweighted adjacency matrix $A$. Define $D = \{D_1, \ldots, D_K\}$, where $D_1, \ldots, D_K$ are $N \times N$ matrices satisfying $[D_k]_{ij} \in \{0, 1\}$ for all $k = 1, \ldots, K$ and $i, j = 1, \ldots, N$, and $\sum_{k=1}^{K} D_k = 1_{N \times N} - I_{N \times N}$. Each matrix $D_k$ presents the pairs of nodes, indicated by ones, for which the probabilities of mislearning the existence of the edges are equal. The probabilities of removing edges in the subsets are given in $\epsilon_1 = \{\epsilon_{11}, \ldots, \epsilon_{K1}\}$, and the probabilities of adding edges are given in $\epsilon_2 = \{\epsilon_{12}, \ldots, \epsilon_{K2}\}$. The graph error model specified by $D$, $\epsilon_1$, and $\epsilon_2$ can then be written as

$$W = A - \sum_{k=1}^{K} \Delta_{\epsilon_{k1}} \odot D_k \odot A + \sum_{k=1}^{K} \Delta_{\epsilon_{k2}} \odot D_k \odot (1_{N \times N} - A). \tag{M3}$$

This construction allows to build very flexible and accurate graph error models, which can adjust to basically any type of graph topology learning errors. However, flexible model requires large $K$, i.e., a lot of parameters. Therefore, it is of interest in practice to only approximate the graph error by using a reasonable value of $K$. The accuracy of such approximation is then appears as trade off with the requirements to the amount and reliability of prior information about the graph structure of a graph signal and the complexity in terms of the value of $K$, i.e., the number of parameters. It is worth noting here that the Erdős-Rényi graph-based error model (M2) is a special case of graph error model (M3). Indeed, it can be obtained by letting $K = 1$ and $D_1 = 1_{N \times N} - I_{N \times N}$.
For further insights into the above introduced generalized graph error model (M3), we take as an example the SBM. Let $C_{k,m}$ denote an $N \times N$ matrix such that $[C_{k,m}]_{i,j} = 1$ if $i \neq j$, $\sum_{i=1}^{k-1} N_i < i \leq \sum_{i=1}^{k} N_i$ and $\sum_{i=1}^{m-1} N_i < j \leq \sum_{i=1}^{m} N_i$, and $[C_{k,m}]_{i,j} = 0$, otherwise. Then we can set, for example, $K = r^2$, $D_1 = C_{1,1}$, $D_2 = C_{1,2}, \ldots, D_{r^2} = C_{r,r}$, and we obtain a model where the probability of mislearning an edge depends on which communities the start and the end nodes belong to. For such model, the following result about the structure of the true adjacency matrix $A$ and that of the mismatched one $W$ as modelled by (M3) is of interest.

**Proposition 3.** Assume that the true adjacency matrix $A$ follows the SBM with $r$ communities, the mismatched adjacency matrix $W$ follows model (M3), and $D_1, \ldots, D_K$ are chosen according to the SBM structure of $A$. Then $W$ retains SBM with the same communities.

**Proof:** When all such pairs of nodes where one node belongs to $i$th community and the other one belongs $j$th community of $A$ are in the same set $D_k$, the probability for edges between those pairs of nodes remains constant in the adjacency matrix $W$ mismatched according to (M3) as well. The latter trivially means that the SBM structure of $A$ is retained in $W$. □

It is also interesting to note that a natural graph error model related to the previously used PPM can be derived using model (M3) by letting $K = 2$, $D_1 = \sum_{k=1}^{r} C_{k,k}$, and $D_2 = 1_{N \times N} - I_{N \times N} - \sum_{k=1}^{r} C_{k,k}$.

**F. Models for Weighted Graphs**

The above defined models can be extended to signals on weighted graphs. Then, in addition to adding and removing edges, we have to consider changes in the edge weights and also figure out how to generate the weights for added edges in a mismatched adjacency matrix.

Let us start with extending graph error model (M2) (extension of graph error model (M1) is similar). Let $\mathcal{A}$ denote the set of nonzero elements of the true graph adjacency matrix $A$. The inaccurately learned weighted adjacency matrix can be then modelled as

$$W = A + (1_{N \times N} - \Delta_{\epsilon_1}) \odot 1_{\mathcal{A}} \odot \Sigma_{\epsilon} - \Delta_{\epsilon_1} \odot A + \Delta_{\epsilon_2} \odot B \odot (1_{N \times N} - 1_{\mathcal{A}}).$$  

(M2w)

Like in graph error model (M2), $\epsilon_1$ is the probability to erroneously removing edges while $\epsilon_2$ is the probability of erroneously adding edges. The second term in (M2w) perturbs the weights of the remaining edges using an $N \times N$ matrix $\Sigma_{\epsilon}$ whose elements are drawn from a zero mean
Gaussian distribution with variance $c \cdot \sigma^2$, where $\sigma^2$ is the sample variance of $A$ and $c$ is the variance multiplier parameter. The third term in (M2w) models the erroneous removal of edges and the fourth term in (M2w) models the erroneous addition of edges with weights given by the matrix $B$ which is an $N \times N$ matrix whose elements are derived by the same rules which were used for edge weights of $A$ if applicable, or alternatively the elements can be drawn from $A$ with replacement. Typically for practical GSP tasks, only positive edge weights are considered, and then, if the above described process produces negative elements in $W$, they should be put to zero.

Similarly, the weighted generalized graph error model can be written as a weighted extension of graph error model (M3), that is,

$$W = A + \sum_{k=1}^{K} \left( (1_{N \times N} - \Delta_{\epsilon_{k1}}) \odot D_k \odot 1_A \odot \Sigma_{ck} - \Delta_{\epsilon_{k1}} \odot A + \Delta_{\epsilon_{k2}} \odot B_k \odot D_k \odot (1_{N \times N} - 1_A) \right).$$

(M3w)

where $\epsilon_{k1}$, $\epsilon_{k2}$ and $D_k$, $k = 1, \ldots, K$ are defined in the same way as in Subsection II-E, and $B_k$ and $\Sigma_{ck}, k = 1, \ldots, K$ are analogous to $B$ and $\Sigma_c$ above.

III. GRAPH AUTOCORRELATION

In GSP tasks such as filtering or signal recovery from samples, smoothness with respect to the chosen shift matrix is a key assumption. Therefore, smoothness is also instrumental in the shift operator learning from training data, as can be also seen for example in [15]. Graph autocorrelation is a measure of smoothness which we will discuss here.

As a model for graph signal we use the moving average model which have been used at least since [7]. Thus, the graph moving average (GMA) signal model of order $m$, denoted hereafter as GMA($m$), is an extension for graph signals of the traditional time series moving average (MA) model, and it is given as

$$z = y + \sum_{i=1}^{m} \theta_i A^i y$$

Later in the paper, we deal with a related concept for multivariate signals in the context of ICA, that is, graph autocorrelation matrix.
where \( y \triangleq [y_1, \ldots, y_N]^T \) with \( y_1, \ldots, y_N \sim N(0, \sigma_y^2) \) being mutually independent Gaussian random variables with zero mean and variance \( \sigma_y^2 \), and \( \theta_1, \ldots, \theta_m \) are MA coefficients. In this paper, we mainly consider the GMA(1) model

\[
    z = y + \theta_l Ay = \tilde{A} y
\]

where \( \tilde{A} = I_{N \times N} + \theta A \).

We start with the autocovariance of a graph signal. For a centered graph signal, its autocovariance at lag \( k \) with respect to \( W \) is defined as

\[
    s_{z,k}(W) = \text{tr}\{\text{cov}(z, W^k z)\} = \frac{1}{N-k} \text{tr}\left\{ \frac{1}{N} z^T W^k z \right\} = \frac{1}{N-k} \mathbb{E}\{ z^T W^k z \}. \tag{4}
\]

Note here that another quadratic form \( z^T Lz \), where \( L \) is the graph Laplacian, appears often in GSP literature, for example in [15]. Since the off-diagonal elements of the Laplacian matrix are the opposites of those of the adjacency matrix and the diagonal elements of the Laplacian are given by the node degrees whereas the diagonal elements of the adjacency matrix are zeros, the two quadratic forms as measures of smoothness are closely related but not fully equivalent. Large graph autocovariance values indicate smoothness, but in the case of the Laplacian, small values indicate smoothness. Indeed, assume that there are two adjacency matrices \( W_1 \) and \( W_2 \), and the matrix \( W_1 \) gives a larger graph autocovariance value for \( z \) and \( k = 1 \) than that of the matrix \( W_2 \). Then for the corresponding Laplacian matrices \( L_1 \) and \( L_2 \), respectively, it holds that \( z^T L_1 z < z^T L_2 z \) any time when autocovariance values are distinct and not close to each other, and only if the autocovariance values are close to each other, then sometimes \( z^T L_1 z > z^T L_2 z \).

For a GMA(1) signal \( z \) from (3), we obtain

\[
    s_{z,k}(W) = \frac{1}{N-k} \mathbb{E}\left\{ y^T \tilde{A}^T W^k \tilde{A} y \right\} = \frac{1}{N-k} \text{tr}\left\{ W^k \mathbb{E}\left\{ \tilde{A} y (\tilde{A} y)^T \right\} \right\} \\
    = \frac{\sigma_y^2}{N-k} \text{tr}\left\{ W^k (I_{N \times N} + \theta A^T)(I_{N \times N} + \theta A) \right\}
\]

where the property \( \text{tr}\{AB\} = \text{tr}\{BA\} \) is used in the first step, and the second step follows from the formula of the covariance matrix \( \mathbb{E}\{zz^T\} \) of \( z = \tilde{A} y \), see [38]. Thus, graph autocovariance for GMA(1) signal is a weighted sum of the covariances between the nodes of a graph signal.

As a standardized version of the graph autocovariance we define graph autocorrelation which is invariant to changes in scales of the graph signal and the matrix \( W \). Then the following definition can be given.

\[\text{September 28, 2020 DRAFT}\]
Definition 1. The graph autocorrelation of lag $k$ with respect to matrix $W$ is

$$r_{z,k}(W) = \frac{\mathbb{E}\{z^\top W^k z\}}{(\mathbb{E}\{|z|^2\} \mathbb{E}\{|W^k z|^2\})^{1/2}}.$$  

It is interesting to see how the graph autocorrelation in Definition 1 depends on the specific parameters of random graph error model, for example, parameters $\epsilon_1$ and $\epsilon_2$ of graph error model (M2). It can be seen that the graph autocorrelation is also a function of the graph adjacency matrix $A$. Thus, let us also assume that $A = \Delta_\alpha$, i.e., $A$ also follows the Erdős-Rényi model with the probability parameter $\alpha$.

For graph autocorrelation $r_{z,1}(W)$ of the GMA(1) signal $z$, we would like to derive the expected value, when both $A$ and $W$ are considered to be random, that is, the expected value $\mathbb{E}_{A,W,z}\{r_{z,1}(W)\}$ for the lag $k = 1$. Even though this expected value is an average over Erdős-Rényi graphs $A$, we have verified by extensive simulations that the graph autocorrelation values are very similar for any specific and fixed realization of $A$. The expected graph autocorrelation value can be then expressed in terms of the essential parameters only, such as the GMA coefficient $\theta$, the probability parameter $\alpha$ in the graph model $A = \Delta_\alpha$, and for example, the parameters $\epsilon_1$ and $\epsilon_2$ in the graph error model given by (M2). The expected value $\mathbb{E}_{A,W,z}\{r_{z,1}(W)\}$ can be used for example to verify, that GMA signal is smoother with respect to its adjacency matrix than the mismatched adjacency matrix.

Example: Expected value of the graph autocorrelation at $k = 1$ for GMA(1) and graph error model (M2).

We approximate the expected value of the ratio in $r_{z,1}(W)$ by ratio of expected values, and the expected value of product in the denominator by the product of expected values.

We start by rescaling the matrices $A_a = aA$ and $W_w = wW$ and choosing the variance $\sigma_y^2$ so that the following approximation holds

$$\mathbb{E}_{A,W,z}\{r_{z,1}(W)\} \approx \mathbb{E}_{A_a,W_w,z}\{z^\top Wz\}.$$  

In order to have this, $a$, $w$ and $\sigma_y^2$ need to be chosen so that the following three equations are

$^5$Even though the elements are not independent, it will be seen in simulations that this approximation is accurate.
satisfied.

\[
\frac{1}{N} \mathbb{E}\{||z||^2\} = 1
\]
\[
\frac{1}{N} \mathbb{E}\{||A_\alpha z||^2\} = 1
\]
\[
\frac{1}{N} \mathbb{E}\{||W_w z||^2\} = 1.
\]

For given values of \(N, \alpha, \epsilon_1\) and \(\epsilon_2\), the only unknowns in (5) are \(a, w\) and \(\sigma_y^2\). Hence we can find \(a, w\) and \(\sigma_y^2\) by simply solving the system of three equations (5). Notice that only the third equation includes \(w\), which implies that \(a\) and \(\sigma_y^2\) can be found from the first two equations. However, the expectations in (5) need to be calculated first. For calculating the expected values in (5), the fact that \(\mathbb{E}\{y_i y_j\} = \sigma_y^2\) if \(i = j\) and \(\mathbb{E}\{y_i y_j\} = 0\) otherwise can be used together with the assumption that the elements of the matrices \(\Delta_\alpha, \Delta_{\epsilon_1},\) and \(\Delta_{\epsilon_2}\) are independent within each matrix as well as between the matrices.

The parameter \(w\) can be found from the last equation in (5) as

\[
w = (\sigma_y^2 (\alpha^3 N^2 \theta^2 a^2 (\epsilon_1 + \epsilon_2 - 1)^2 + \alpha^2 N^2 \theta^2 a^2 (1 - \epsilon_1 + \epsilon_2 (2(\epsilon_1 + \epsilon_2) - 3)))
\]
\[
\quad + \alpha N^2 \theta^2 a^2 (\epsilon_2 - \epsilon_1^2) + \alpha^2 N (\theta^2 a^2 (\epsilon_1 + \epsilon_2 - 1) - (\epsilon_1 + \epsilon_2 - 1)^2)
\]
\[
\quad + \alpha N (1 - \epsilon_1 + \epsilon_2 (2(\epsilon_1 + \epsilon_2) - \theta^2 a^2 - 3)) + N (\epsilon_2 - \epsilon_1^2) + \alpha (\epsilon_1 + \epsilon_2 - 1) - \epsilon_2)^{-1/2}
\]

and the parameters \(a\) and \(\sigma_y^2\) can be found by solving the system of remaining two equations that after substituting the above found \(w\) becomes

\[
\begin{align*}
\sigma_y^2 ((\alpha - \alpha^3) \theta^2 a^2 N - 2 \alpha \theta a + 1) &= 1 \\
\sigma_y^2 ((\alpha^2 - \alpha^3) \theta^2 a^4 N^2 - 2 \alpha^2 \theta a^3 N + (\alpha - \alpha^2) a^2 N) &= 1.
\end{align*}
\]

Finally, we find for \(W\) from (M2) with parameters \(\epsilon_1\) and \(\epsilon_2\), the closed-form approximate expression for the expected autocorrelation value of the graph signal \(z\) as a function of \(\epsilon_1\) and \(\epsilon_2\) as

\[
\mathbb{E}_{A, W, z}\{r_{z,1}(W)\} \approx \sigma_y^2 (\theta a w N (\alpha - \alpha^2) - \alpha w) (1 - \epsilon_1 - \epsilon_2) - w \epsilon_2.
\]

The analysis of (6) is non-trivial because, for example, \(w\) depends on all other parameters. Thus, we leave it to a numerical study in the next section.

\footnote{For more details, see Appendix B}
IV. NUMERICAL STUDY

A. Study Case 1: Graph Autocorrelation of GMA(1) Signal for the Mismatched Adjacency Matrix

In our first example, we examine how the expected graph autocorrelation of the GMA(1) signal derived in (6) changes when the matrix with respect to which it is computed is gradually changed from the adjacency matrix of the signal according to graph error model (M2). We assume that \( z \) is a GMA(1) signal with an unweighted Erdös-Rényi adjacency matrix \( A \) with probability \( \alpha \) and coefficient \( \theta \) as parameters.

Figs. 1 and 2 illustrate the behavior of the graph autocorrelation as a function of \( \epsilon_1 \) and \( \epsilon_2 \), respectively, when one of them varies and the other one is kept fixed. Both theoretical values and averages of 2000 graph autocorrelations from simulated datasets are shown. For each run, a new Erdös-Rényi adjacency matrix \( A \) is generated with parameters \( N = 500 \) and \( \alpha = 0.05 \), and the GMA coefficient is \( \theta = 0.5 \). The choice of the variance parameter value \( \sigma^2_y \) has no effect on the results because of the standardization in the graph autocorrelation (see the definition).

The figures show that the graph autocorrelation is decreasing with respect to both \( \epsilon_1 \) and \( \epsilon_2 \), which suggests that graph autocorrelation is useful for estimating the adjacency matrix even if we only have a single realization of the graph signal.

B. Study Case 2: Graph Error Effect on GMA Graph Filter

As in any other GSP task, the adjacency matrix has a key role in graph filtering. In fact, any linear and shift-invariant filter (commutes with other shift-invariant filters) can be written as a polynomial of the form \([1]\)

\[
F_{\text{GMA}}(W) = h_0 I_{N \times N} + h_1 W + \cdots + h_K W^K
\]

where \( K \) is smaller than or equal to the degree of the minimal polynomial of \( W \).

The coefficients \( h_0, \ldots, h_K \) can be found by using least squares for solving the system of equations

\[
\begin{align*}
  h_0 + h_1 \lambda_1 + \cdots + h_K \lambda^K_1 &= \alpha_1 \\
  \vdots \\
  h_0 + h_1 \lambda_N + \cdots + h_K \lambda^K_N &= \alpha_N
\end{align*}
\]
where $\lambda_1, \ldots, \lambda_N$ are eigenvalues of $W$ ordered in increasing order of their distance to the maximum magnitude, i.e., $d_n = |\max\{|\lambda_1|, \ldots, |\lambda_N|\} - \lambda_n|$. Then $\lambda_1, \ldots, \lambda_N$ can be interpreted as graph frequencies from lowest to highest [39], and $\alpha_1, \ldots, \alpha_N$ are the corresponding frequency responses.

In this example, we study the filter sensitivity to the choice of the adjacency matrix, when the high-pass filter is defined by the following frequency responses

$$\alpha_n = \begin{cases} 
1, & \text{if } d_n > \text{median}\{d_1, \ldots, d_N\} \\
0, & \text{otherwise}
\end{cases}$$

and it is applied to detecting malfunctioning sensors as in [39]. The data are the daily temperatures in the year 2016 from 150 Finnish weather stations [40]. Stations that had at most one missing
Fig. 2. Theoretical (solid) and simulated (dash) values of graph autocorrelations of order $k = 1$. The lines from top to bottom are for $\epsilon_1 = 0, 0.1, 0.2, 0.3$.

observation were selected, and since those missing observations were from the same day in November, after dropping that day out, we have a clean data of 365 days (leap year). The technique to detect the outlying measurements, presented in [39], is to high-pass filter the data and threshold the Fourier transform coefficients of the output by the maximum absolute value of the GFT coefficients from the three previous days. If any of the coefficients exceed the threshold value, it is diagnosed that at least one of the sensors is not working properly.

In this example, there is no obvious choice of a benchmark adjacency matrix\textsuperscript{7} but we choose

\textsuperscript{7}In other example, there always exists an obvious choice for benchmark adjacency matrix.
a weighted 6-nearest neighbors graph like in [39] with edge weights

\[ a_{kl} = \frac{e^{-(d_{kl}/20)^2}}{\sqrt{\sum_{j \in N_k} e^{-(d_{kj}/20)^2} \sum_{j \in N_l} e^{-(d_{lj}/20)^2}}} \]  

where \( d_{kl} \) is the distance between the locations of the \( k \)th and \( l \)th sensors in kilometers. We use the general graph error model because it is not sensible to connect two stations that are far away from each other. We set the threshold to 250 kilometers above which a pair of stations is treated as no longer connected. Otherwise, the probabilities of removing and creating connections have constant values \( \epsilon_1 \) and \( \epsilon_2 \) between the pairs. Hence, in graph error model (M3w), which is then applicable here, we have the matrices

\[
[D_1]_{kl} = \begin{cases} 
1, & \text{if } d_{kl} \leq 250 \\
0, & \text{otherwise}
\end{cases}
\]

\[
[D_2]_{kl} = \begin{cases} 
1, & \text{if } d_{kl} > 250 \\
0, & \text{otherwise}
\end{cases}
\]

and probabilities \( \epsilon_{11} = \epsilon_1, \epsilon_{12} = \epsilon_2, \epsilon_{21} = \epsilon_2, \epsilon_{22} = 0 \).

By perturbing the edge weights, we deviate slightly from graph error model (M3w), so that the variance of the Gaussian component related to incoming edges of a given node is the variance multiplier \( c \) times the variance of the incoming edge weights of that node in the 6-nearest neighbors graph, not the variance of the edge weights of the whole 6-nearest neighbors graph. We do not allow negative weights, but use zero weights instead.

In the experiment, we change one sensor value at a time to +20 Celsius degrees and perform the test described above using adjacency matrices given by all combinations of edge removing probabilities \( \epsilon_1 = 0, 0.01, 0.02, 0.03 \), edge adding probabilities \( \epsilon_2 = 0, 0.1, 0.2, 0.3 \), and variance parameter values \( c = 0, 0.005, 0.01, 0.015 \).

Finding the malfunctioning sensors can be seen as a binary classification task. Theoretically, all classifiers (the same method with different graphs) have equal proportion of false positives of about 25\%, because the maximum GFT coefficient today is larger than the maximum of the GFT coefficients from the three previous days one out of four times when there are no outliers. Hence, we can focus on the accuracy in finding the true positives. The benchmark adjacency matrix given by \( \epsilon_1 = \epsilon_2 = c = 0 \) found the malfunctioning sensors with 84\% accuracy. For the other combinations, 200 realizations of the adjacency matrix are generated.
In Fig. 3 we give the averages for different values of parameters of interest when other parameters are zeros. There are no visible patterns of interactions between the parameters, i.e., the shapes of the curves look identical at all levels of the other two parameters. The results show that adding edges to the graph has no effect, but removing edges weakens the performance. Also, the perturbation of the weights has negative impact, though the reduction is significant only for values from 0 to 0.005, and for \( c = 0.005, 0.01, 0.015 \), the numbers are approximately the same. This suggests that the number of edges in the 6-nearest neighbors graph might be too small for these purposes, but that the edge weights are good as they are, because small changes in them worsen the performance. The reason why addition of edges has smaller effect can be partly explained by the fact that they have smaller edge weight than the edges in the original nearest neighbors graph.

C. Study Case 3: Graph Error Effect on Graph Autoregressive Moving Average Filter

First order graph autoregressive moving average (GARMA) filter \([41]\) has coefficients \( \phi, \psi \) and \( c \), and uses some graph Laplacian matrix \( L \). We use here the translated normalized Laplacian

\[
L = -T^{-1/2}WT^{-1/2}
\]

where \( T \) is a diagonal matrix of node degrees and \( W \) is a symmetric adjacency matrix. Note that the eigenvalues of \( L \) are in the interval \([-1, 1]\), but as in the GMA filter case, they can be very different for the cases of the true \( A \) and the estimated \( W \) adjacency matrices.

The output of GARMA(1) filter is given by \( z = y + cx \), where \( x \) is the input and \( y \) is the state after convergence of the of recursion \( y_{t+1} = \phi Ly_t + \psi x \). GARMA filter of order \( K \) can be constructed from \( K \) parallel GARMA(1) filters, whose coefficients are designed based on the eigenvalues of \( L \).

Again, the errors in the adjacency matrix \( W \) have a significant effect on the filter directly via \( L \), but also via distorting the spectrum of \( L \). The errors then can reflect even on the GARMA filter stability.

We are following a simulation setup in \([25]\). Undirected and unweighted graphs are created by generating \( N = 100 \) random points on the area \([0, 1] \times [0, 1]\) using the uniform distribution, and connecting two points if the distance between them is less than \( 0.15\sqrt{2} \).
Let $L = \mathbf{V} \Lambda \mathbf{V}^\top$ be the eigendecomposition of the translated normalized graph Laplacian, and let $\lambda_n$ and $v_n$ denote the $n$th diagonal element of $\Lambda$ and the $n$th column vector of $\mathbf{V}$, i.e., the $n$th eigenvalue and eigenvector, respectively. The eigenvalues are in the interval $[-1, 1]$. Then the graph signal is given by $x = \bar{x} + n$, where $\bar{x}$ is a low frequency signal satisfying $\bar{x}^\top v_n = 1$, if $\lambda_n < 0$, and $\bar{x}^\top v_n = 0$, otherwise, and $n$ is Gaussian noise with zero mean and covariance matrix $0.1 \mathbf{I}_{N \times N}$.

Frequency responses of GARMA graph filters of orders $K = 1, 3, 5, 7$ are designed to match the frequency content of the signal $\bar{x}$. Outputs of the filters, denoted by $z^{(e)}$, are compared to the output of the ideal filter, denoted by $z^{(d)}$. The performance is then measured using the square
root of the mean square error

$$\sigma_e = \left[ \text{tr}(ee^T)/N \right]^{1/2}$$

where $e = z^{(e)} - z^{(d)}$.

The values in Figs. 4 and 5 are averages over 2000 runs for each pair $(\epsilon_1, \epsilon_2)$ in graph error model (M2) that is applicable here. The results show that higher order GARMA filters are more accurate when the correct adjacency matrix is used. On the other hand, the lower order filters are more robust to graph errors, and thus all filters perform almost equally when the error probabilities grow.

Fig. 4. Averages of $\sigma_e$ over 2000 repetitions for GARMA filter orders $K = 1, 3, 5, 7$ and probabilities $\epsilon_1$ in graph error model (M2), when $\epsilon_2 = 0$. 
Fig. 5. Averages of $\sigma_e$ over 2000 repetitions for GARMA filter orders $K = 1, 3, 5, 7$ and probabilities $\epsilon_2$ in graph error model [M2], when $\epsilon_1 = 0$.

D. Study Case 4: Graph Error Effect on ICA of Graph Signals

In this example, we use both numerical study and theoretical results to investigate the effect of non-optimal signal graph knowledge to the performance of GraDe [7], [42] for graph signal separation in ICA problem.

Let $X \in \mathbb{R}^{P \times N}$ denote centered $P$-dimensional graph signal generated as a mixture of independent components according to the model [42]

$$X = \Omega Z$$

where $\Omega \in \mathbb{R}^{P \times P}$ is a full rank mixing matrix, $Z \in \mathbb{R}^{P \times N}$ is the matrix of $P$ mutually independent graph signals with zero means and unit variances.
The goal of ICA is to estimate the unmixing matrix $\Gamma = \Omega^{-1}$ using only the signal matrix $X$.

Let $X_w = \hat{S}_0^{-1/2}X$ be the whitened signals, where $\hat{S}_0$ is the sample covariance matrix of $X$. In GraDe, the unmixing matrix estimate is obtained by diagonalizing/jointly diagonalizing one or more graph autocorrelation matrices given as

$$\hat{S}_k(W) = \frac{1}{N - k}(X_w W^k X_w^T), \quad k = 1, \ldots, K$$

i.e., by finding the orthogonal $U$ which maximizes the objective function

$$\sum_{k=1}^{K} ||\text{diag}(US_k(W)U^T)||^2.$$  

The unmixing matrix estimate is then given as $\hat{\Gamma} = US_0^{-1/2}$. A fast algorithm for the joint diagonalization is available in [43] and it is applicable for the case when the shift matrix $W$ is chosen to be symmetric, or the graph-autocorrelation matrices are symmetrized. The unmixing matrix estimate for an inaccurately learned adjacency matrix $W$ is denoted as $\hat{\Gamma}(W)$. Notice that GraDe reduces to the well-known second-order blind identification (SOBI) estimator [44], when $W$ is the cyclic graph.

We will use the following (see [45] for details) asymptotic result, derived in the context of the SOBI estimator, for an unmixing matrix estimate $\hat{\Gamma}$ obtained using joint diagonalization of matrices $\hat{S}_1, \ldots, \hat{S}_K$. When $\Omega = I_{P \times P}$, for $i \neq j$, we have

$$\sqrt{N} (\hat{\gamma}_{ii} - 1) = -\frac{1}{2} \sqrt{N} ([\hat{S}_0]_{ii} - 1) + o_p(1)$$

and

$$\sqrt{N} \hat{\gamma}_{ij} = \frac{\sum_k (\lambda_{ki} - \lambda_{kj}) (\sqrt{N} [S_k]_{ij} - \lambda_{ki} \sqrt{N} [S_0]_{ij})}{\sum_k (\lambda_{ki} - \lambda_{kj})^2} + o_p(1)$$

(8)

where $\lambda_{ki} \triangleq E\{[S_k]_{ii}\}$, and $o_p(1)$ stands for negligible terms. The diagonal elements of $\hat{\Gamma}$ do not depend asymptotically on $\hat{S}_1, \ldots, \hat{S}_K$, and thus, in the case of graph signals ICA, do not depend on $W$. Therefore, the sum of variances of the off-diagonal elements

$$\text{SOV}(\hat{\Gamma}(W)) = N \sum_{j \neq i} \text{var}(\hat{\Gamma}(W)_{ij})$$

(9)

can be used when comparing the separation efficiencies induced by different choices of $W$. We will use the ratio of the sums given as

$$R(W_1, W_2) = \frac{\text{SOV}(\hat{\Gamma}(W_1))}{\text{SOV}(\hat{\Gamma}(W_2))}.$$
Particularly, $W_1$ is the adjacency matrix which is used for generating the source components and $W_2$ is a perturbed version of the adjacency matrix. Using the ratio of the sum of variances instead of the sums helps to visualize the efficiency losses.

We consider ICA model where the independent components are GMA(1) signals with symmetric and unweighted adjacency matrix $A$, and evaluate the performance of the GraDe estimate with $K = 1$ and the matrix $W$, which is a mismatched version $A$ obtained according to graph error model (M2) for undirected graphs. In this model, the asymptotic variances of $\hat{\Gamma}(W)$, which are needed for computing $SOV(\hat{\Gamma}(W))$, can be calculated using (8). Expressions for these asymptotic variances can be derived in the same way as the corresponding formulas for the variances in the time series context in [45].

The performance in the numerical simulations is measured using the minimum distance (MD) index

$$D(\hat{\Gamma}) \triangleq \frac{1}{\sqrt{P-1}} \inf_{C \in \mathcal{C}} \| C \hat{\Gamma} \Omega - I_{P \times P} \|$$

where $\mathcal{C} \triangleq \{ C : \text{each row and column of } C \text{ has exactly one non-zero element} \}$. The MD index takes values between zero and one, and it is invariant with respect to the mixing matrix. Also, there is a connection between the minimum distance index and the sum of variances of the off-diagonal elements when $\Omega = I_{P \times P}$, given as

$$N(P-1)E\{D(\hat{\Gamma})^2\} \to SOV(\hat{\Gamma}), \text{ as } N \to \infty$$

(10)

where SOV is defined in (9).

For two sets of estimates, $W_1$ and $W_2$, we define

$$\hat{R}(W_1, W_2) = \frac{\text{ave}\{D(\hat{\Gamma}(W_1))^2\}}{\text{ave}\{D(\hat{\Gamma}(W_2))^2\}}$$

where the averages are found over 1000 Monte Carlo trials. Equation (10) implies that $\hat{R}(W_1, W_2) \approx R(W_1, W_2)$ for large $N$.

Erdős–Rényi matrices $A = \Delta_\alpha$ with different values of $\alpha$ are used as the adjacency matrices of GMA signals. The estimate $\hat{\Gamma}(A)$ (with true $A$) is a natural benchmark to which we compare the estimates obtained using $W$.  

\footnote{Indeed, the extension of asymptotic variance expression in [45] from the time series context to the context of graph signals is straightforward, but the formula appears to be too long to be presented here because of the space limitation.}
TABLE I

$R(A, W)$ for $A$ with $\alpha = 0.05$ and $W$ given by $\epsilon_1 = 0, 0.1, \ldots, 0.5$ and $\epsilon_2 = 0, 0.01, \ldots, 0.05$.

| $\epsilon_1 \backslash \epsilon_2$ | 0 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|----------------------------------|---|------|------|------|------|------|
| 0                               | 1.00 | 0.81 | 0.68 | 0.58 | 0.51 | 0.45 |
| 0.1                             | 0.88 | 0.71 | 0.58 | 0.49 | 0.43 | 0.38 |
| 0.2                             | 0.77 | 0.60 | 0.49 | 0.41 | 0.35 | 0.30 |
| 0.3                             | 0.66 | 0.50 | 0.40 | 0.33 | 0.28 | 0.24 |
| 0.4                             | 0.56 | 0.41 | 0.31 | 0.26 | 0.21 | 0.18 |
| 0.5                             | 0.46 | 0.31 | 0.24 | 0.19 | 0.15 | 0.13 |

TABLE II

$\hat{R}(A, W)$ from 1000 repetitions for $\alpha = 0.05$ and $\epsilon_1 = 0, 0.1, \ldots, 0.5$ and $\epsilon_2 = 0, 0.01, \ldots, 0.05$.

| $\epsilon_1 \backslash \epsilon_2$ | 0 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 |
|----------------------------------|---|------|------|------|------|------|
| 0                               | 1.00 | 0.81 | 0.67 | 0.56 | 0.48 | 0.44 |
| 0.1                             | 0.88 | 0.65 | 0.56 | 0.47 | 0.43 | 0.35 |
| 0.2                             | 0.76 | 0.59 | 0.46 | 0.40 | 0.34 | 0.29 |
| 0.3                             | 0.62 | 0.46 | 0.37 | 0.32 | 0.27 | 0.25 |
| 0.4                             | 0.52 | 0.37 | 0.29 | 0.25 | 0.21 | 0.20 |
| 0.5                             | 0.43 | 0.31 | 0.25 | 0.20 | 0.17 | 0.16 |

In Tables I and II, the values of $R(A, W)$ and $\hat{R}(A, W)$ are shown, respectively, when $A$ is $1000 \times 1000$ matrix with $\alpha = 0.05$ and there are $p = 4$ independent components generated using (2) with $\theta = 0, 0.2, 0.4, \text{and } 0.6$. For Table II, we generate 1000 datasets for each pair $(\epsilon_1, \epsilon_2)$ and always generate a new $W$. In Table I, the sum of variances is an average for ten $W$’s, even though $\text{SOV}(\hat{\Gamma}(W))$ is quite stable for fixed $\epsilon_1$ and $\epsilon_2$. The simulation results match the theoretical values quite well. When looking at the results with respect to error probabilities $\epsilon_1$ and $\epsilon_2$, it seems that GraDe is more sensitive to adding irrelevant edges than missing the real edges. However, notice that when $\alpha = 0.05$, then in numbers of mislearned edges $\epsilon_2 = 0.01$ corresponds to $\epsilon_1 = 0.19$.

For four selected pairs $(\epsilon_1, \epsilon_2)$, Fig. 6 plots $R(A, W)$ as a function of $\alpha$ that is used in creating $A$. The curves display the averages of ten values given by different $W$’s. As expected, the efficiency loss caused by inaccuracy in the adjacency matrix is the larger, the more sparse the graph is.
V. CONCLUSION AND DISCUSSION

In this paper, the effect of graph adjacency matrix mismatch has been analyzed and graph error models for different types of graphs, e.g., directed/undirected and weighted/unweighted graphs, have been developed. The complexity of the error models varies from simple Erdős-Rényi type model to one which captures nonconstant edge mislearning probabilities. The latter is of interest because it has been reported in GSP literature that deleting one edge can have immensely larger effect than deleting another edge. Better understanding and formalization of what kind of edges are important or why connecting some pairs of nodes is more harmful than others is therefore crucial. The graph error models have been applied for studying graph error effects in graph signal filtering and graph signal ICA applications, where both theoretical arguments and numerical studies for real and synthetic data have been used. The results for
different application examples differed in whether missing or extra links were more detrimental, and in the GARMA filter example, it was observed that the higher-order filters were more sensitive to graph errors. These findings suggest that the graph error effects need to be studied case by case, and that competing GSP methods may differ in terms of their robustness to graph errors. Therefore, studying the robustness properties of the GSP methods as well as developing robust GSP methods is of high importance for GSP in general.

APPENDIX A

PROOF OF THEOREM 1

A. Introduction to Graphons

Limits of sequences (when the number of nodes tends to infinity) of dense graphs (such graphs in which each node is connected to positive percent of the other nodes) cannot be easily defined using graphs with countable set of nodes since there is no uniform distribution on countably infinite sets. Hence, uncountable cardinality of nodes is needed in order to have a natural graph sequence limit object. In this context graphon $W$ is interpreted as a weighted graph with uncountable set of nodes, $W(x, y)$ giving the edge weight between nodes $x$ and $y$. In [47], centrality measures of graphons were defined and they were shown to be limit objects of centrality measures of finite size graphs.

Another role of graphons is a kernel in the kernel graph model, which consists of the triple $G = (N, W, \mu)$, where $N$ is the number of nodes, $W$ is a graphon (or kernel) and $\mu$ is a function/mapping from a probability space to the space $[0, 1]^N$.

A realization of a kernel graph is obtained by generating values $u_1, \ldots, u_N$ from $\mu$. Then the $i$th and the $j$th nodes are connected with probability $W(u_i, u_j)$ independently from the other pairs. Thus, the graphon can be seen here as sort of a density function for the graph structure. Typically $\mu$ is chosen to have uniform density on $[0, 1]^N$.

Some variants of the kernel graph model have been discussed in the literature. For example, $N$ can be derived through a random process or $\mu = \mu_D$ can be taken as a deterministic function that gives sequence $u_i = i/N$ for $i = 1, \ldots, N$ with probability one. For example, the Erdős-Rényi model with probability parameter $\epsilon$, which is of interest here, is obtained by choosing the constant graphon $W(x, y) = \epsilon$ for all $0 \leq x, y \leq 1$ and for any choice of $\mu$. 

September 28, 2020 DRAFT
The SBM with fixed community sizes \( N_1, \ldots, N_r \) is given by the deterministic \( \mu_D \) and \( W = W_{SB} \) satisfying \( W_{SB}(x, y) = p_{k,l} \) when \( \sum_{i=1}^{k-1} N_i / N < x \leq \sum_{i=1}^{k} N_i / N \) and \( \sum_{i=1}^{l-1} N_i / N < y \leq \sum_{i=1}^{l} N_i / N \) where \( \sum_{i=1}^{0} N_i = 0 \). When \( \mu \) has the uniform distribution and \( W = W_{SB} \), we have a SBM with random community sizes, the expected sizes being \( N_1, \ldots, N_r \).

In the kernel graph model, the expected degree of a random node is
\[
(N - 1) \int_0^1 \int_0^1 W(x, y) d\mu(x) d\mu(y)
\]
and the range of expected degrees is
\[
(N - 1) \int_0^1 W(x, y) d\mu(y), \ x \in [0, 1].
\]

While the stochastic block model allows for a very heterogeneous degree distribution with the extreme setup \( r = N \) and \( N_1 = \cdots = N_N = 1 \), such construction is quite cumbersome. Therefore, it is of interest to use continuous graphons which only have few parameters, such as the exponential model \[33\] where the graphon is of the form
\[
W(x, y) = e^{-(\beta_1(x+y)+\beta_0)}
\]
with \( \beta_0, \beta_1 \geq 0 \). Parameter \( \beta_0 \) controls here the sparsity of the graph. The larger the value of \( \beta_0 \) is, the more sparse the graph is. Moreover, increasing the parameter \( \beta_1 \) makes the degree distribution more heterogeneous.

The eigenvalues of large normalized and unweighted graph adjacency matrices can be then approximated in terms of the eigenvalues and eigenfunctions of the kernel \[35\]. The eigenvalues \( \lambda \) and eigenfunctions of the kernel corresponding to graphon \( W \) can be found by solving the equation
\[
\lambda f(x) = \int_0^1 W(x, y) f(y) dy.
\]

Due to symmetry, any eigenvalue \( \lambda \) is real-valued. Examples of solving the eigenvalue/eigenfunction problem in the cases of Erdős-Rényi, stochastic block and exponential models can be found in \[33\], where the behavior of the eigenvalues of finite graphs is also studied in terms of simulations. Thus, graphons have been proved to be useful in the analysis of graph sequences, but they are also convenient for formulating other theoretical results. Here, we are rather interested to justify the use of Erdős-Rényi graphon as a basic model for modelling graph errors in graph signals.
B. Proof of Theorem 7

Consider the random variable \( c_1 W_1(x, y) \) where \( x \) and \( y \) are fixed and \( W_1 \) is picked randomly from \( \mathcal{W}_c \). First notice that the distribution is the same for all pairs \((x, y)\), and therefore \( \mathbb{E}\{c_1 W_1(x, y)\} = \mathbb{E}\{c_1\} \mathbb{E}\{W_1(x, y)\} = c \epsilon \). The result then follows by applying the law of large numbers.

**APPENDIX B**

**DERIVATION OF EXPECTED VALUE OF THE GRAPH AUTOCORRELATION FOR THE EXAMPLE IN SECTION III**

Let us first compute here the expectation in the first equation in (5). The calculations of the expectations in the other two equations are explained afterwards shortly. We start by substituting (2) into the expectation in the first equation in (5), and opening the square

\[
\mathbb{E}\{\|z\|^2\} = \mathbb{E}\{\| (\theta A y + y) \|^2 \} \\
= \mathbb{E}\left\{ \sum_{i=1}^{N} \left( \theta \sum_{k=1}^{N} a_{ik} y_k + y_i \right) - \frac{1}{N} \sum_{j=1}^{N} \left( \theta \sum_{l=1}^{N} a_{jl} y_l - y_j \right) \right\}^2 \\
= \mathbb{E}\left\{ \sum_{i=1}^{N} \left( \theta \sum_{k=1}^{N} a_{ik} y_k + y_i \right) - \frac{1}{N} \sum_{j=1}^{N} \left( \theta \sum_{l=1}^{N} a_{jl} y_l - y_j \right) \right\} \times \left( \theta \sum_{k'=1}^{N} a_{ik} y'_k + y_i - \frac{1}{N} \sum_{j'=1}^{N} \left( \theta \sum_{l'=1}^{N} a_{ij'} y'_l - y'_j \right) \right). 
\]

Then after some algebraic manipulations of the latter, we obtain that

\[
\mathbb{E}\{\|z\|^2\} = \mathbb{E}\left\{ \sum_{i=1}^{N} \left( \theta^2 \sum_{k=1}^{N} a_{ik}^2 y_k^2 - \frac{2}{N} \sum_{j=1}^{N} \left( \theta a_{j'i} y_i^2 - y_i y_j \right) \right) + \frac{\theta^2}{N^2} \sum_{j=1}^{N} a_{j'i} a_{ij} y_i^2 \\
+ \frac{2\theta}{N^2} \sum_{j=1}^{N} a_{j'i} y_i^2 + \frac{1}{N^2} \sum_{j=1}^{N} y_j^2 \right\} \\
= \sigma_y^2 \left( N^2 \theta^2 \alpha a^2 - 2N \theta^2 \alpha a^2 - 2N^2 \theta^2 \alpha^2 a^2 - 2N \theta \alpha a + N \right. \\
- 2N \theta \alpha a - 2 + N \theta^2 \alpha a^2 + N^2 \theta^2 \alpha^2 a^2 + 2N \theta \alpha a + 1 \right) .
\]
Dropping the negligible terms in the above expression, i.e., the zero-order terms of $N$ and first-order terms of $N$ that also include $a^2$, the first equation in (5) can be approximated as

$$\frac{1}{N} \mathbb{E}\{\|z\|^2\} \approx \sigma_y^2 (\alpha - \alpha^2) \theta^2 a^2 N - 2\alpha \theta a + 1.$$ 

For deriving the expected values in the other two equations of (5), there are sums over terms of the type $w_{ij} w_{i'j'} a_{kl} a_{k'l'} y_m^2$. To calculate the expected values of them, we can use the fact that $w_{ij}$ and $a_{kl}$ are independent if $\min(i, j) \neq \min(k, l)$ or $\max(i, j) \neq \max(k, l)$, and then

$$\mathbb{P}(w_{ij} = w \text{ and } a_{kl} = a) = \mathbb{P}(w_{ij} = w) \mathbb{P}(a_{kl} = a) = (\alpha(1 - \epsilon_1) + (1 - \alpha)\epsilon_2) \alpha.$$ 

If $\min(i, j) = \min(k, l)$ and $\max(i, j) = \max(k, l)$, we have

$$\mathbb{P}(w_{ij} = w \text{ and } a_{kl} = a) = (1 - \epsilon_1) \alpha.$$ 

REFERENCES

[1] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Jul. 2013.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Process. Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[3] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proceedings of the IEEE*, vol. 106, no. 5, 808–828, May 2018.

[4] S. Y. Tu and A. H. Sayed, “Mobile adaptive networks,” *IEEE J. Select. Topics Signal Process.*, vol. 5, no. 4, pp. 649–664, Apr. 2011.

[5] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, “Opinion fluctuations and disagreement in social networks,” *Mathematics of Operations Research*, vol. 38, no. 1, pp. 1–27, Jan. 2013.

[6] W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, “Graph frequency analysis of brain signals,” *IEEE J. Select. Topics Signal Process.*, vol. 10, no. 7, pp. 1189–1203, Oct. 2016.

[7] F. Blöchl, A. Kowarsch, and F. J. Theis, “Second-order source separation based on prior knowledge realized in a graph model,” in “*Latent Variable Analysis and Signal Separation,* LNCS (V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, and E. Vincent, eds.), vol. 6365, (Heidelberg), pp. 434–441, Springer, 2010.

[8] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, “Discrete signal processing on graphs: Sampling theory,” *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[9] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Sampling of graph signals with successive local aggregations,” *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832–1843, Apr. 2016.

[10] 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.

[11] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, “Reconstruction of graph signals through percolation from seeding nodes,” *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4363–4378, Aug. 2016.
[12] A. Sandryhaila, S. Kar, and J. M. Moura, “Finite-time distributed consensus through graph filters,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Florence, Italy, May 2014, pp. 1080–1084.

[13] S. Safavi and U. A. Khan, “Revisiting finite-time distributed algorithms via successive nulling of eigenvalues,” IEEE Signal Process. Letters, vol. 22, no. 1, pp. 54–57, Jan. 2015.

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

[15] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, “Learning sparse graphs under smoothness prior,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., New Orleans, Louisiana, USA, Mar. 2017, pp. 6508–6512.

[16] P. A. Traganitis Y. Shen, and G.B. Giannakis, “Network topology inference via elastic net structural equation models,” in Proc. European Signal Processing Conference, Kos island, Greece, Aug.–Sept. 2017, pp. 156–160.

[17] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G.Rabbat, “Characterization and inference of graph diffusion processes from observations of stationary signals,” IEEE Trans. Signal and Information Processing over Networks, vol. 4, no. 3, pp. 481–496, Sept. 2018.

[18] S. Segarra, S., A. G. Marques, G. Mateos, and A. Ribeiro, “Network topology inference from spectral templates,” IEEE Trans. Signal and Information Processing over Networks, vol. 3, no. 3, pp. 467–483, Sept. 2017.

[19] E. Ceci and S. Barbarossa, “Graph signal processing in the presence of topology uncertainties,” IEEE Trans. Signal Process., vol. 68, pp. 1558–1573, Feb. 2020.

[20] E. Ceci, Y. Shen, G. B. Giannakis and S. Barbarossa, “Graph-based learning under perturbations via total least-squares,” IEEE Trans. Signal Process., vol. 68, pp. 2870–2882, Feb. 2020.

[21] F. Gama, J. Bruna, and A. Ribeiro, “Stability properties of graph neural networks”, arXiv preprint arXiv:1905.04497 Sep. 2019.

[22] 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.

[23] S. Kruzick and J. M. Moura, “Optimal filter design for signal processing on random graphs: Accelerated consensus,” IEEE Trans. Signal Process., vol. 66, no. 5, pp. 1258–1272, Mar. 2017.

[24] S. Segarra, and A. Ribeiro, “Stability and continuity of centrality measures in weighted graphs,” IEEE Trans. Signal Process., vol. 64, no. 3, pp. 543–555, Feb. 2016.

[25] E. Isufi, A. Loukas, A. Simonetto and G. Leus, “Filtering random graph processes over random time-varying graphs,” IEEE Trans. Signal Process., vol. 65, no. 16, pp. 4406–4421, Aug. 2017.

[26] E. Isufi, A. Loukas, A. Simonetto and G. Leus, “On the transferability of spectral graph filters,” in Proc. Int. Conf. Sampling Theory and Applications, IEEE, Bordeaux, France, July 2019, pp. 1–5.

[27] T. Cai, T. Liang, and A. Rakhlin, “On detection and structural reconstruction of small-world random networks,” IEEE Trans. Network Science and Engineering, vol. 4, no. 3, pp. 165–176, 2017.

[28] E. Abbe, A. S. Bandeira, A. Bracher, and A. Singer, “Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery,” IEEE Trans. Network Science and Engineering, vol. 1, no. 1, 10–22, 2014.

[29] S.A. Vorobyov, A.B. Grshman, and Z.-Q. Luo, “Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem,” IEEE Trans. Signal Process., vol. 51, no. 2, pp. 313–324, Feb. 2003.

[30] J. Miettinen, S.A. Vorobyov, and E. Ollila, “Graph error effect in graph signal processing,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Calgary, Canada, Apr. 2018, pp. 4164–4168.

[31] P. Erdős and A. Rényi, “On random graphs,” Publ. Math. Debrecen, vol. 6, pp. 290–297, 1959.
[32] P.W. Holland, K. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” *Social Networks*, vol. 5, no. 2, pp. 109–137, Jun. 1983.

[33] M. W. Morency and G. Leus, “Signal processing on kernel-based random graphs,” in *Proc. European Signal Processing Conference*, Kos Island, Greece, Aug.–Sept. 2017, pp. 385–389.

[34] L. Lovász and B. Szegedy, “Limits of dense graph sequences,” *Journal of Combinatorial Theory B*, vol. 96, pp. 933–957, 2006.

[35] L. Lovász, *Large networks and graph limits*. Providence: American Mathematical Society, vol. 60, 2012.

[36] B. Bollobás, S. Janson and O. Riordan, “The phase transition in inhomogeneous random graphs,” *Random Structures and Algorithms*, vol. 31, no. 1, pp. 3–122, Jan. 2007.

[37] K. Avrachenkov, L. Cottatellucci, L. and A. Kadavankandy, “Spectral properties of random matrices for stochastic block model,” in *Proc. IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2015, pp. 537–544.

[38] S. P. Chepuri and G. Leus, “Graph sampling for covariance estimation,” *IEEE Trans. Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 451–466, Sept. 2017.

[39] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Frequency analysis,” *IEEE Trans. Signal Processing*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[40] *National Climatic Data Center*, [ftp://ftp.ncdc.noaa.gov/pub/data/gsod/], 2016.

[41] E. Isufi, A. Loukas, A. Simonetto and G. Leus, “Autoregressive moving average graph filtering,” *IEEE Trans. Signal Processing*, vol. 65, no. 2, pp. 274–288, Jan. 2017.

[42] J. Miettinen, E. Nitzan, S. A. Vorobyov and E. Ollila, “Graph signal processing meets blind source separation”, arXiv preprint [arXiv:2008.10461] Aug. 2020.

[43] D. Clarkson, “Remark AS R74: A least squares version of algorithm AS 211: The F-G diagonalization algorithm,” *Applied Statistics*, vol. 37, pp. 317–321, 1988.

[44] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines, “A blind source separation technique using second-order statistics,” *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, Feb. 1997.

[45] J. Miettinen, K. Illner, K. Nordhausen, H. Oja, S. Taskinen, and F. Theis, “Separation of uncorrelated stationary time series using autocovariance matrices,” *Journal of Time Series Analysis*, vol. 37, no. 3, pp. 337–354, Mar. 2016.

[46] P. Ilmonen, K. Nordhausen, H. Oja, and E. Ollila, “A new performance index for ICA: Properties computation and asymptotic analysis,” in “*Latent Variable Analysis and Signal Separation*,” *LNCS* (V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, and E. Vincent, eds.), vol. 63–65, (Heidelberg), pp. 229–236, Springer, 2010.

[47] M. Avella-Medina, F. Parise, M. Schaub, and S. Segarra, “Centrality measures for graphons: Accounting for uncertainty in networks,” *IEEE Trans. Network Science and Engineering*, vol. 7, no. 1, pp. 520–537, 2018.