Graph signal separation based on smoothness or sparsity in the frequency domain

Sara Mohammadi, Massoud Babaie-Zadeh, Dorina Thanou

***Abstract*—In this paper, we study the problem of demixing an observed signal, which is the summation of a set of signals that live on a multi-layer graph, by proposing several methods to decompose the observed signal into structured components. For this purpose, we build on two of the most widely-used graph signal models' assumptions, namely smoothness and sparsity in the graph spectral domain. We firstly show that a vector can be *uniquely* decomposed as the summation of a set of smooth graph signals, up to the indeterminacy of their DC values. So, if the original signals are known to be smooth, it is expected that with such a decomposition all of the original signals are retrieved. From the blind source separation (BSS) point of view, this is like the separation of a set of graph signals from a single mixture, contrary to traditional BSS in which at least two observed mixtures are typically required. Then, the approach is generalized to a wider family of graph signals, which are not necessarily smooth, but exhibit some sparse frequency characteristics in the graph spectral domain. Numerical simulations confirm the good performance of our approach in separating a mixture of graph signals.**

***Index Terms*—graph signal separation, blind source separation, graph signal processing, multi-layer graphs.**

I. Introduction

IN COMPLEX structures, such as social and sensor networks, relationship between the data can be described by a graph. In the emerging field of graph signal processing (GSP) [1], [2], graphs are used as prior information to mitigate data analysis, and many classical digital signal processing concepts have been generalized to graph signals [1], [2]. In more complicated structures like transportation [3] and biological [4], [5] networks, multiple types of relationships can exist between data samples. In these cases, multi-layer graphs that consist of several graph layers with the same nodes but different connectivity patterns are appropriate tools for encoding various forms of relations [6].

On the other hand, blind source separation (BSS) [7] is a field in signal processing in which the goal is to retrieve a set of source signals from a set of their mixtures without having any information about the mixing process or the original sources (hence the term '*blind*'). BSS problems have various applications in different domains, *e.g.* image processing and communications [7]. In BSS, the number of observed (mixed) signals is typically more than one, and in many papers, the number of observed signals is assumed to be greater than or equal to the number of sources.

S. Mohammadi and M. Babaie-Zadeh are with the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran (e-mail: 9810sara.m@gmail.com, mbzadeh@yahoo.com).

D. Thanou is with the Center for Intelligent Systems (CIS), EPFL, Lausanne, Switzerland (e-mail: dorina.thanou@epfl.ch).

Since the emergence of GSP, some researchers have tried to extend BSS methods to graph signals. In [8], [9], [10], [11], [12], [13] a second order BSS approach (based on joint diagonalization of graph autocovariance or autocorrelation matrices) is generalized to graph signals. In their approach, as in many traditional BSS papers, the number of the observed signals is assumed to be equal to the number of sources.

In this work, we consider the following graph signal separation problem: Let $\mathbf{x}_1, \dots, \mathbf{x}_K \in \mathbb{R}^N$ be graph signals over separate layers of a known K-layer graph and $\mathbf{x} \triangleq \sum_{i=1}^{K} \mathbf{x}_i$. Graph layers shown by $\mathcal{G}_1, \dots, \mathcal{G}_K$ are known, undirected, and connected graphs that have the same nodes but are different in their connectivity patterns. The layer signals $\mathbf{x}_i, i = 1, \dots, K$ are not known, and only their summation, *i.e.* $\mathbf{x}$, is observed. The goal is then to retrieve $\mathbf{x}_1, \dots, \mathbf{x}_K$ from the observed signal $\mathbf{x}$, probably by having extra assumptions on $\mathbf{x}_i$'s, for example, their smoothness or their sparsity in the frequency domain. This problem is actually a BSS problem of graph signals from only one observed mixture.

To further clarify the problem, consider the following illustrative example. Suppose that the number of infections of an epidemic disease such as COVID-19 in N cities is known and is denoted by $\mathbf{x} \in \mathbb{R}^N$. These cities lay on K different known transportation networks (flights, trains, etc), each of which has N nodes and can be considered as a layer of a multi-layer graph. The number of infections over different cities that spread through the transportation network i can be considered as an (unknown) graph signal $\mathbf{x}_i$ residing on the nodes of i-th graph layer. The total observed number of infections, $\mathbf{x}$, can then be attributed to the sum of infections captured at each graph layer, *i.e.*, $\mathbf{x}_1, \dots, \mathbf{x}_K$. These layer graph signals are unknown, and only the total number of infections in each city, $\mathbf{x}$, is observed. The goal is then to retrieve the original graph signals $\mathbf{x}_1, \dots, \mathbf{x}_K$ from $\mathbf{x}$, and understand the contribution of each transportation network to the total number of infections. In this problem, the unknown layer signals ($\mathbf{x}_i$'s) can be assumed to be smooth on their layer graph, because if two cities are heavily connected on a transportation network, it is expected that the infections due to this transportation network in these two cities are relatively close.

In this paper, we firstly show that, under some mild assumptions and indeterminacies, $\mathbf{x}$ can be *uniquely* decomposed into the summation of a set of smooth graph layer signals $\mathbf{x}_i$. So, if the original layer signals $\mathbf{x}_i$ are known to be smooth graph signals, it is expected that with such a decomposition, all of the original signal components are retrieved. Note that no statistical independence between the components is assumed; only their smoothness is exploited. Moreover, in the second part of the paper, even the smoothness assumption is relaxed

and replaced either with the assumption that the frequency supports, *i.e.* the indices of non-zero frequency components, of the graph signals are known a priori, or with the assumption that all graph signals are sparse in the frequency domain with nearly similar frequency supports.

From the BSS point of view, this consists in separating a set of source signals by having only one mixture of them. Note that our approach (and the uniqueness of decomposing a signal into a set of smooth signals) has no counterpart in traditional digital signal processing. Actually, although traditional digital signals are special cases of graph signals, our approach cannot be used for the separation of traditional digital signals from only one mixture, because traditional digital signals reside all on *the same graph*. However, here we are implicitly exploiting the difference of graphs on which the underlying smooth signals reside.

Up to our best knowledge, there is only one paper [14] that addresses the demixing of multiple graph signals from only one mixture, but for very specific source signals: In our work, the graph layer signals are assumed to be smooth or sparse in the frequency domain, while the main assumption of [14] is that each graph layer signal is the result of the diffusion of an originally sparse input through that layer via an unknown linear graph filter with a known order. Then, the authors of [14] propose an approach to separate the individual graph signals by estimating their unknown input supports and the coefficients of the diffusing graph filters.

The rest of the paper is organized as follows. In Section II, a few necessary preliminaries from GSP are very briefly reviewed. Section III is devoted to the proposed approach for separating smooth graph signal from noiseless and noisy mixtures. Some enhancements for retrieving spectrally constrained graph signals are then proposed in Section IV. Finally, in Section V, numerical results are provided.

II. GSP Background

In this section, some GSP concepts are very briefly reviewed from [1]. An undirected weighted graph is defined by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent the set of N nodes, and the set of edges, respectively. Moreover, $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a weighted symmetric adjacency matrix showing the weights between each pair of nodes. Diagonal entries of $\mathbf{W}$ are all equal to zero (*i.e.* there is no self loop). The graph Laplacian matrix is defined as $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$, where the degree matrix $\mathbf{D}$ is diagonal, and $d_{ii} = \sum_{j=1}^{N} w_{ij},\ \forall i = 1, \ldots, N$. It is proved that $\mathbf{L}$ is a positive semidefinite matrix. The eigenvalue decomposition of the Laplacian matrix is represented as $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N]$ is an orthonormal matrix including eigenvectors, and $\mathbf{\Lambda}$ is a diagonal matrix of nonnegative eigenvalues, $0 = \lambda_1 \leq \ldots \leq \lambda_N$. Note $\lambda_1 = 0$ and its corresponding eigenvector is a constant vector (*i.e.* the all-one vector normalized to unit-norm). Moreover, for a connected graph, only λ_1 is equal to zero.

A graph signal $\mathbf{x}$, is a real vector whose i-th entry, x_i, shows a value residing on the i-th node of $\mathcal{G}$. In GSP, the concept of classical Fourier transform is extended to Graph Fourier Transform (GFT). For a graph signal $\mathbf{x}$, GFT and its inverse are defined as $\hat{\mathbf{x}} \triangleq \mathbf{V}^T\mathbf{x}$ and $\mathbf{x} \triangleq \mathbf{V}\hat{\mathbf{x}}$, respectively.

The variation of a signal on a graph, also known as smoothness of a graph signal $\mathbf{x}$, is measured by the following graph Laplacian quadratic form

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i,j} w_{ij}(x_i - x_j)^2 = \sum_{i=1}^{N} \lambda_i \hat{x}_i^2, \qquad (1)$$

where $\hat{x}_i$'s are the graph Fourier coefficients of $\mathbf{x}$ (*i.e.* the entries of $\hat{\mathbf{x}}$). The first equality in (1) shows that for smoother graph signals, the value of $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is smaller.

In this paper, the average of a graph signal $\mathbf{x}$ is denoted by $\bar{x}$, so, for a graph signal with N nodes, $\bar{x} \triangleq \frac{1}{N}\sum_{i=1}^{N} x_i$. Note also that $\bar{x} = \frac{1}{\sqrt{N}}\hat{x}_1$, where $\hat{x}_1$ is the first graph Fourier coefficient of $\mathbf{x}$. In this paper, both $\bar{x}$ and $\hat{x}_1$ are interchangeably referred to as the "DC of $\mathbf{x}$".

A multi-layer graph consists of multiple graph layers that show different relations between data samples [6]. In this paper, the i-th graph layer of a K-layer graph is represented by a weighted and undirected graph $\mathcal{G}_i = (\mathcal{V}, \mathcal{E}_i, \mathbf{W}_i),\ \forall i = 1, \ldots, K$, where $\mathcal{V}$ is the vertex set including N nodes and is the same for all graph layers, and $\mathcal{E}_i$ and $\mathbf{W}_i$ indicate the edge set and the weighted symmetric adjacency matrix of the i-th graph layer, respectively. In this setting, the graph signal over the i-th graph layer is denoted by $\mathbf{x}_i$, and $\bar{x}_i$ shows its DC value.

III. Decomposing an observed signal into the summation of smooth graph signals

In this section, we make the assumption that the observed signal $\mathbf{x}$ is generated from the summation of K graph signals, each of which is a smooth signal on a layer of a multi-layer graph. Our goal is then to recover the graph signals $\mathbf{x}_1, \ldots, \mathbf{x}_K$ from the observed signal $\mathbf{x}$, by exploiting the smoothness of each unknown signal component on the corresponding layer. In the following, two different cases are addressed separately: exact decomposition, which is suitable for decomposing a noiseless mixture of graph signals, and approximate decomposition, which is suitable for decomposing a noisy mixture of graph signals.

A. Exact decomposition

Let $\mathbf{L}_i$ be the Laplacian matrix of $\mathcal{G}_i$, and so $\mathbf{x}_i^T\mathbf{L}_i\mathbf{x}_i$ measures the smoothness of $\mathbf{x}_i$ on $\mathcal{G}_i$. The goal of the *exact* decomposition is to decompose $\mathbf{x}$ as $\mathbf{x} = \mathbf{x}_1 + \cdots + \mathbf{x}_K$ such that all $\mathbf{x}_i$'s are smooth on their layers in the multi-layer graph. To do this, the following quadratic programming problem is proposed

$$\underset{\{\mathbf{x}_i\}_{i=1}^K}{\text{minimize}} \quad \sum_{i=1}^{K} \mathbf{x}_i^T\mathbf{L}_i\mathbf{x}_i \quad \text{s.t.} \quad \mathbf{x} = \sum_{i=1}^{K} \mathbf{x}_i. \qquad (2)$$

In the following theorem, we show that, under some specific constraints, the above problem has a unique solution.

Theorem 1. *If all graph layers $\{\mathcal{G}_i\}_{i=1}^K$ are connected graphs, problem* (2) *has a unique solution up to DC values of $\mathbf{x}_i$'s.*

Proof: See Appendix A. ∎

The above theorem states that in (2) the DC values of $\mathbf{x}_i$'s are free parameters as long as $\sum_{i=1}^{K} \bar{x}_i = \bar{x}$. Hence, in order to get a unique solution, we ignore the DC values of $\mathbf{x}_i$'s by assuming that they are equal to zero, and remove the DC of $\mathbf{x}$ by defining $\mathbf{z} \triangleq \mathbf{x} - \bar{x}\mathbf{1}$. In this way, the following modified form of (2) has a unique solution:

$$\underset{\{\mathbf{x}_i\}_{i=1}^{K}}{\text{minimize}} \quad \sum_{i=1}^{K} \mathbf{x}_i^T \mathbf{L}_i \mathbf{x}_i \quad \text{s.t.} \begin{cases} \mathbf{z} = \sum_{i=1}^{K} \mathbf{x}_i \ , \\ \bar{x}_i = 0 \ , \quad \forall i = 1, \ldots, K. \end{cases} \tag{3}$$

B. Approximate decomposition

Next, we relax the exact decomposition assumption, with an approximate decomposition $\mathbf{x} \approx \sum_{i=1}^{K} \mathbf{x}_i$. So, (2) is reformulated as

$$\underset{\{\mathbf{x}_i\}_{i=1}^{K}}{\text{minimize}} \quad \|\mathbf{x} - \sum_{i=1}^{K} \mathbf{x}_i\|_2^2 + \sum_{i=1}^{K} \gamma_i \mathbf{x}_i^T \mathbf{L}_i \mathbf{x}_i \ , \tag{4}$$

where γ_i's are regularization parameters determining the relative cost of the approximation and the smoothness of the resulting graph signals.

Theorem 2. *If all graph layers $\{\mathcal{G}_i\}_{i=1}^{K}$ are connected graphs, then problem* (4) *has a unique solution up to DC values of $\mathbf{x}_i$'s.*

Proof: See Appendix B. ∎

Therefore, similar to (3), problem (4) is modified as follows, which has a unique solution:

$$\begin{aligned} \underset{\{\mathbf{x}_i\}_{i=1}^{K}}{\text{minimize}} \quad & \|\mathbf{z} - \sum_{i=1}^{K} \mathbf{x}_i\|_2^2 + \sum_{i=1}^{K} \gamma_i \mathbf{x}_i^T \mathbf{L}_i \mathbf{x}_i \\ \text{s.t.} \quad & \bar{x}_i = 0 \ , \quad \forall i = 1, \ldots, K. \end{aligned} \tag{5}$$

Note that optimization problems (3) and (5) are Quadratic Programming (QP) problems, so they can easily be solved by using convex optimization tools [15].

C. Discussions and modifications

To gain a better insight into the approaches of Sections III-A and III-B, we firstly state (3) in the frequency domain. The GFT of $\mathbf{x}_i, \ \forall i = 1, \ldots, K$ is $\hat{\mathbf{x}}_i = \mathbf{V}_i^T \mathbf{x}_i$, where $\mathbf{V}_i = [\mathbf{v}_{i1}, \ldots, \mathbf{v}_{iN}]$ is the matrix of the eigenvectors of $\mathbf{L}_i$, and $\mathbf{V}_i \mathbf{V}_i^T = \mathbf{I}$. So, replacing $\mathbf{x}_i$ in (3) with $\mathbf{V}_i \hat{\mathbf{x}}_i$ results in

$$\underset{\{\hat{\mathbf{x}}_i\}_{i=1}^{K}}{\text{minimize}} \quad \sum_{i=1}^{K} \sum_{j=2}^{N} \lambda_{ij} \hat{x}_{ij}^2 \quad \text{s.t.} \begin{cases} \mathbf{z} = \sum_{i=1}^{K} \mathbf{V}_i \hat{\mathbf{x}}_i \ , \\ \hat{x}_{i1} = 0 \ , \quad \forall i = 1, \ldots, K, \end{cases} \tag{6}$$

where $\hat{\mathbf{x}}_i = [\hat{x}_{i1}, \ldots, \hat{x}_{iN}]^T$ and λ_{ij} is the j-th eigenvalue of the $\mathbf{L}_i$.

The above equation reveals a drawback of the decomposition approach of (3), which happens when the range and/or distribution of eigenvalues of different $\mathbf{L}_i$'s are highly different. For example, suppose that there are two graphs: For $\mathcal{G}_1$, $\lambda_2 = 5.0$, $\lambda_3 = 10.0$ and for $\mathcal{G}_2$, $\lambda_2 = 1.0$, $\lambda_3 = 1.5$. Since the third eigenvalue of $\mathcal{G}_2$ is less than the second eigenvalue of $\mathcal{G}_1$, (6) prefers to put more energy on the third frequency component of $\mathcal{G}_2$ rather than putting it on the second frequency (smoothest) component of $\mathcal{G}_1$, which is probably not what is sought in many applications. This will actually be verified in the simulation results of Section V-B.

To overcome this drawback, an idea is that instead of λ_{ij}'s in (6), some other weights are used in such a way that the algorithm has the tendency to use lower frequency components of all graphs. Replacing $\lambda_{ij}, \ \forall i = 1, \ldots, K$ in (6) with w_j leads to

$$\underset{\{\hat{\mathbf{x}}_i\}_{i=1}^{K}}{\text{minimize}} \quad \sum_{j=2}^{N} w_j \sum_{i=1}^{K} \hat{x}_{ij}^2 \quad \text{s.t.} \begin{cases} \mathbf{z} = \sum_{i=1}^{K} \mathbf{V}_i \hat{\mathbf{x}}_i \ , \\ \hat{x}_{i1} = 0 \ , \quad \forall i = 1, \ldots, K, \end{cases} \tag{7}$$

where $w_j, \ \forall j = 2, \ldots, N$ are user-provided weight parameters such that $0 < w_2 < w_3 < \ldots < w_N$. Note that w_j represents the relative cost of assigning the signal to the j-th frequency of all graphs, independently from the values of λ_{ij}. So, (7) has the tendency to firstly use lower frequency components of all graphs, before using higher frequencies, independently of the spectrum of the graphs.

Theorem 3. *Problem* (7) *has a unique solution.*

Proof: See Appendix C. ∎

Remark: An exactly similar modification can be proposed for (5). However, to avoid repetition it is not explained here.

IV. Decomposing an observed signal into the summation of spectrally constrained graph signals

Smoothness of a graph signal means that Fourier coefficients of the signal corresponding to lower frequencies have significantly higher values than other Fourier coefficients. The decomposition approach of the previous section can be easily extended to the case where the graph signals have only a few non-zero Fourier coefficients but not necessarily corresponding to lower frequencies (so not necessarily smooth). In this section, this extension is done for two cases: where the location of the non-zero coefficients (*i.e.* the frequency support) is a priori known, and where it is not known.

A. Known frequency support

When the frequency support of the graph signals are known, (7) can directly be used with the difference that instead of $w_2 < w_3 < \ldots < w_N$, the values of w_j's should be set such that they are smaller for the frequency components which are known to be non-zero, and highly larger for other frequency components.

B. Unknown frequency support

When the frequency support of the graph signals is not a priori known, the decomposition $\mathbf{x} = \sum_{i=1}^{K} \mathbf{x}_i$ can be done in a way that the total number of frequency components that are jointly equal to zero is maximized. In other words, by defining the matrix $\hat{\mathbf{X}} \triangleq [\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_K]$, the goal is that as many as possible *rows* of $\hat{\mathbf{X}}$ are equal to zero. So, if $\hat{\mathbf{x}}_{[j]}^T$ stands for the j-th *row* of $\hat{\mathbf{X}}$, the ℓ_0 pseudo-norm (*i.e.* the number of non-zero entries) of the vector $\hat{\mathbf{x}}_{\ell_2} \triangleq [\|\hat{\mathbf{x}}_{[1]}\|_2, \|\hat{\mathbf{x}}_{[2]}\|_2, \ldots, \|\hat{\mathbf{x}}_{[N]}\|_2]^T$

is to be minimized, which leads to the following optimization problem:

$$\underset{\{\hat{\mathbf{x}}_i\}_{i=1}^K}{\text{minimize}} \quad \|\hat{\mathbf{x}}_{\ell_2}\|_0 \quad \text{s.t.} \begin{cases} \mathbf{z} = \sum_{i=1}^K \mathbf{V}_i \hat{\mathbf{x}}_i \ , \\ \hat{x}_{i1} = 0, \ \forall i = 1, \dots, K. \end{cases} \tag{8}$$

We define $\hat{\mathbf{x}}_{\text{B}} \triangleq [\hat{\mathbf{x}}_{[1]}^T, \dots, \hat{\mathbf{x}}_{[N]}^T]^T$ and $\mathbf{V}_{\text{B}} \triangleq [\mathbf{V}_{[1]}, \dots, \mathbf{V}_{[N]}]$, where the subscript "B" stands for "Block", $\mathbf{V}_{[j]} \triangleq [\mathbf{v}_{1j}, \dots, \mathbf{v}_{Kj}]$ and $\mathbf{v}_{ij}$ is the j-th eigenvector of $\mathbf{L}_i$, so the first condition of (8) simplifies to

$$\mathbf{z} = \sum_{i=1}^K \mathbf{V}_i \hat{\mathbf{x}}_i = \sum_{j=1}^N \mathbf{V}_{[j]} \hat{\mathbf{x}}_{[j]} = \mathbf{V}_{\text{B}} \hat{\mathbf{x}}_{\text{B}}. \tag{9}$$

$\hat{\mathbf{x}}_{\text{B}}$ is called a block k-sparse vector if $\|\hat{\mathbf{x}}_{\ell_2}\|_0 \leq k$. We assume that $\hat{\mathbf{x}}_{\text{B}}^*$ is a block k-sparse vector that satisfies the above equality[1]. This answer is unique if and only if $\mathbf{V}_{\text{B}}\mathbf{c} \neq \mathbf{0}$ for every block $2k$-sparse vector $\mathbf{c} \neq \mathbf{0}$ [16, Proposition 1]. Under this condition, the solution to the problem (8) is unique.

Since problem (8) is not convex, finding its optimal solution is difficult, and would requires a combinatorial search. Therefore, instead of minimizing the ℓ_0 of $\hat{\mathbf{x}}_{\ell_2}$, as in the compressed sensing literature [17], its ℓ_1 is going to be minimized, which results in the convex problem:

$$\underset{\{\hat{\mathbf{x}}_i\}_{i=1}^K}{\text{minimize}} \quad \|\hat{\mathbf{x}}_{\ell_2}\|_1 \quad \text{s.t.} \begin{cases} \mathbf{z} = \mathbf{V}_{\text{B}} \hat{\mathbf{x}}_{\text{B}} \ , \\ \hat{x}_{i1} = 0, \quad \forall i = 1, \dots, K. \end{cases} \tag{10}$$

The uniqueness of the solution of (10) has been established in [16] under some constraints (refer to [16, Theorem 1]).

Remark 1: The advantage of problem (10) over (7) is that problem (10) has no weight parameters to be set manually. However, usage of this problem is only for the case that the graph signals are sufficiently sparse in the frequency domain with nearly similar frequency supports.

Remark 2: Exactly similar extensions can be proposed for (5). However, to avoid repetition they are not explained here.

V. Simulation Results

In this section, the performance of the proposed methods is assessed on seven different scenarios. In all experiments, the Graph Signal Processing Toolbox (GSPBox) implemented in Python (PyGSP) [18] is used to create different graphs. Moreover, CVXPY [19], [20] is utilized to solve all the optimization problems.

A. Experiment 1: Evaluating the performance of (3) *for graphs with similar distribution of eigenvalues*

In this experiment, the performance of (3) is checked over random connected sensor graphs [21]. The reason for using sensor graphs is that they roughly have similar distribution of eigenvalues. To create smooth signals on these graphs, $\mathbf{x}_i$'s are set equal to a linear combination of the second and third eigenvectors of $\mathbf{L}_i$ with coefficients chosen randomly from 0 to 1 (with a uniform distribution). Then, each $\mathbf{x}_i$ is normalized to have entries between -1 and 1, and $\mathbf{x}$ is set equal to $\sum_{i=1}^K \mathbf{x}_i$. Finally, this generated signal is given to (3) to perform graph signal separation, and to hopefully recover the original $\mathbf{x}_i$'s.

As a measure of quality of the recovery of each original signal, we use signal-to-noise ratio (SNR) in dB defined as $\text{SNR}_i \triangleq 10 \log_{10}\big(\|\mathbf{x}_i\|_2^2 / \|\mathbf{x}_i - \mathbf{x}_{\text{est}_i}\|_2^2\big), \ \forall i = 1, \dots, K$, where $\mathbf{x}_{\text{est}_i}$ is the estimated graph signal over the i-th graph. Moreover, $\text{SNR}_{\text{av}} \triangleq \frac{1}{K}\sum_{i=1}^K \text{SNR}_i$ is used as the measure of quality of the whole recovery. Furthermore, to report the qualities, the whole simulation is repeated 200 times, and the values of resulting SNR_i's and SNR_{av}'s are averaged across all simulations (denoted by $\overline{\text{SNR}}_i$ and $\overline{\text{SNR}}_{\text{av}}$). In each simulation, new sets of graphs and graph signals are created.

Table I shows $\overline{\text{SNR}}_i$'s averaged over 200 different experiments, with $K = 5$ and different numbers of nodes denoted by N. As indicated by the high SNR value for each component, by solving (3), we are able to recover the original signals to a large extent.

TABLE I: Performance (in dB) of (3) over $K = 5$ random sensor graphs, averaged over 200 simulations.

N	$\overline{\text{SNR}}_1$	$\overline{\text{SNR}}_2$	$\overline{\text{SNR}}_3$	$\overline{\text{SNR}}_4$	$\overline{\text{SNR}}_5$	$\overline{\text{SNR}}_{\text{av}}$
100	14.96	14.90	14.64	14.93	15.07	14.90
250	21.99	22.03	22.42	22.03	22.15	22.12
450	27.25	26.86	26.94	27.20	27.25	27.10

B. Experiment 2: Evaluating the performance of (3) *for graphs with different distribution of eigenvalues*

In this experiment, the performance of (3) is assessed over *different graph types (families)* in order to create graphs with different distribution of eigenvalues. To generate a synthetic signal $\mathbf{x}$, five random connected graphs are created. The type of each graph is selected randomly from the following graph generative models: *Barabasi–Albert* [22], *random regular* [23], *sensor*, *community* [24], *Erdos Renyi* [25]. Then, $\mathbf{x}_i \ , \forall i = 1, \dots, 5$ and $\mathbf{x}$ are created similar to Experiment 1. Finally, the generated signal is given to (3) to perform graph signal separation. Table II shows averaged results over 200 different simulations for signals with different numbers of nodes (N). In each simulation, new sets of graphs and graph signals are created.

TABLE II: Performance (in dB) of (3) over $K = 5$ different graph types, averaged over 200 simulations.

N	$\overline{\text{SNR}}_1$	$\overline{\text{SNR}}_2$	$\overline{\text{SNR}}_3$	$\overline{\text{SNR}}_4$	$\overline{\text{SNR}}_5$	$\overline{\text{SNR}}_{\text{av}}$
100	7.08	7.91	7.18	7.47	6.81	7.29
250	8.54	9.35	8.76	8.77	8.11	8.71
450	10.27	11.73	11.00	10.91	10.08	10.80

Comparing the results of Tables I and II shows that in this case (different graph types), the performance of (3) has been decreased.

C. Experiment 3: Evaluating the performance of (7) *for smooth graph signals*

In this experiment, the performance of (7) is evaluated for smooth graph signals with different numbers of non-

[1]Note that $\|\hat{\mathbf{x}}_{\ell_2}\|_0 \leq k$ is a stronger assumption than k-sparsity of each graph signal in the frequency domain. Thus, minimizing the number of non-zero rows of $\hat{\mathbf{X}}$ can result in sparser graph signals in the frequency domain.

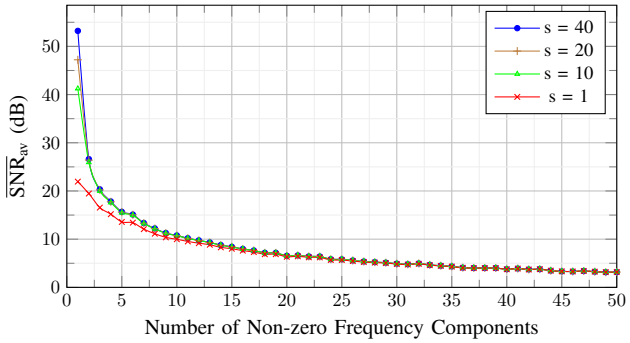


Fig. 1: Performance of methods (7) for noiseless smooth graph signals with respect to the number of non-zero frequency components.

zero frequency components. To generate a synthetic signal $\mathbf{x}$, similar to Experiment 2, five random connected graphs with $N = 250$ nodes are created. Each $\mathbf{x}_i$, $\forall i = 1, \ldots, 5$ with j non-zero frequency components is set equal to the random linear combination of the first j eigenvectors of its Laplacian matrix, without considering the constant eigenvector. Then, each $\mathbf{x}_i$ is normalized to have entries between -1 and 1, and $\mathbf{x}$ is defined as $\sum_{i=1}^{5} \mathbf{x}_i$. For different numbers of non-zero frequency components (j), (7) is used to perform graph signal separation on $\mathbf{x}$. In (7), w_i is set equal to $(i-2) \times s + 1$, $\forall i = 2, \ldots, N$, where s indicates the step size. Figure 1 shows $\overline{\text{SNR}}_{\text{av}}$ for different values of s versus the number of non-zero frequency components (j) averaged over 20 simulations. In each simulation and for each value of j, a new set of graphs is created. As the values of $\overline{\text{SNR}}_{\text{av}}$ in this figure indicate, when the goal is to find smooth graph signals, the performance declines rapidly by increasing the number of non-zero frequency components.

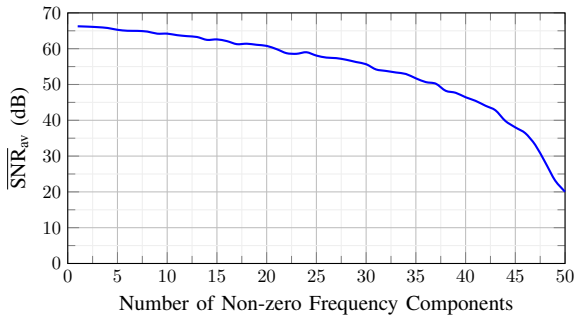
By setting $j = 2$, for different values of N, the performance of (7) with $s = 40$ is tested. Table III shows the results averaged over 200 simulations. Comparing Tables II and III shows the superiority of the performance of (7) over (3), when the graph types are different.

TABLE III: Performance (in dB) of (7) with $s = 40$ over $K = 5$ different graph types, averaged over 200 simulations.

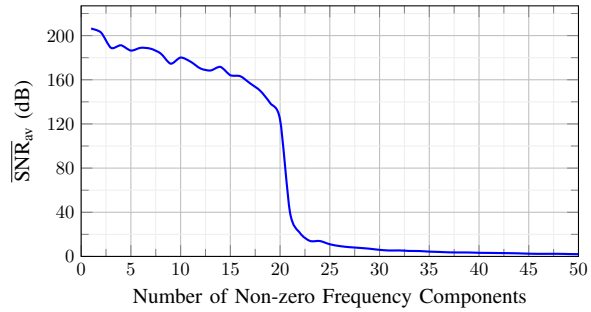
N	$\overline{\text{SNR}}_1$	$\overline{\text{SNR}}_2$	$\overline{\text{SNR}}_3$	$\overline{\text{SNR}}_4$	$\overline{\text{SNR}}_5$	$\overline{\text{SNR}}_{\text{av}}$
100	19.11	19.38	18.89	18.99	18.52	18.98
250	25.52	26.06	25.59	25.43	25.05	25.53
450	29.69	30.37	29.87	29.67	29.08	29.74

D. Experiment 4: Evaluating the performances of (7) *and* (10) *for spectrally constrained graph signals*

In this experiment, the performances of (7) and (10) are evaluated for spectrally constrained graph signals with different numbers of non-zero frequency components. To generate a synthetic signal $\mathbf{x}$, similar to Experiment 2, five random connected graphs with $N = 250$ nodes are created. Each graph $\mathcal{G}_i$ is then used to generate a graph signal $\mathbf{x}_i$, with j non-zero frequency components, as a linear combination of the j eigenvectors of the corresponding Laplacian matrix, with



(a) Known frequency support



(b) Unknown frequency support

Fig. 2: Performance of methods (7) and (10) for noiseless spectrally constrained graph signals with respect to the number of non-zero frequency components.

random weights. These eigenvectors are selected randomly in each simulation, without considering the constant eigenvector. However, we impose that the selection of the precise indices is the same across all graphs (*e.g.* if the second eigenvector is selected, it is selected for all graphs, and hence, all graphs have the same frequency band). Then, each $\mathbf{x}_i$ is normalized to have entries between -1 and 1, and $\mathbf{x}$ is set equal to $\sum_{i=1}^{5} \mathbf{x}_i$. For different numbers of non-zero frequency components (j), (7) and (10) are used independently to perform graph signal separation on $\mathbf{x}$. In (7), the weights corresponding to the j known frequency components are set equal to 1, and the rest are set equal to 10000. Figure 2 shows $\overline{\text{SNR}}_{\text{av}}$ versus the number of non-zero frequency components (j) averaged over 20 simulations. In each simulation and for each value of j, a new set of graphs is created, and new random columns of eigenvector matrices are selected. As the values of $\overline{\text{SNR}}_{\text{av}}$ in this figure indicate, both approaches have been quite successful in recovering the original signals. Moreover, Fig. 2b indicates that by increasing j, a phase transition occurs, that is, the performance declines abruptly when the number of non-zero components passes a threshold. This bears some similarities with a well-known phenomenon from the compressed sensing literature [17]: the uniqueness of the sparse solution of an underdetermined system of linear equations and the equivalence of minimizing ℓ_0 and ℓ_1 norms break out beyond certain sparsities.

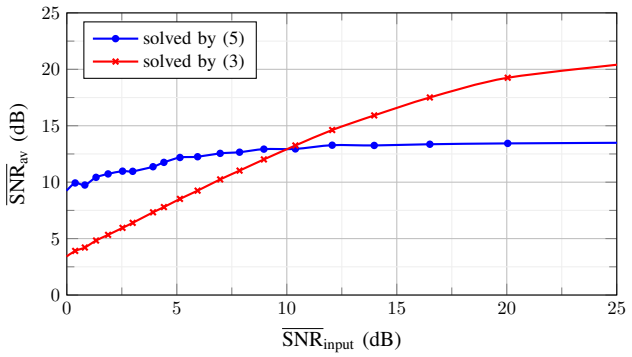


Fig. 3: Average quality of the resulted signals from the approximate decomposition with respect to the quality of the corrupted signal.

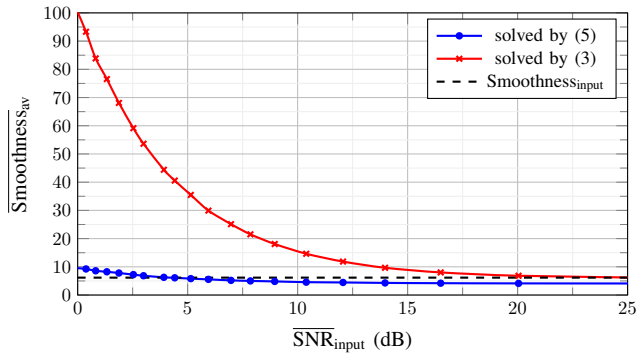


Fig. 4: Average smoothness value of the resulted signals from the approximate decomposition with respect to the quality of the corrupted signal.

E. Experiment 5: Robustness against noise

In this experiment, the robustness of the algorithms against noise is studied. Since enhancements of the noise robust approach (5) (as had been done for (3) to propose (7) and (10)) have not been explicitly stated in the paper, only the performances of (3) and (5) are going to be compared here. To generate a synthetic signal $\mathbf{x}$, four different connected sensor graphs with $N = 250$ nodes are constructed (note that the reason for choosing this graph type is to avoid the mentioned problem in Experiment 2). Then, after initializing $\mathbf{x}$ similar to Experiment 1, we corrupt the signal by adding some white Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, where $\mathbf{I}$ is an identity matrix, and σ^2 is the variance of the additive noise. For the corrupted signal, both (3) and (5) are utilized independently to perform graph signal separation. In (5), $\{\gamma_i\}_{i=1}^4$ are set equal to 2, and the $\text{SNR}_{\text{input}}$ and the $\text{Smoothness}_{\text{av}}$ are computed as follows: $\text{SNR}_{\text{input}} \triangleq 10\log_{10}\big(\|\mathbf{x}-\mathbf{n}\|_2^2/\|\mathbf{n}\|_2^2\big)$ and $\text{Smoothness}_{\text{av}} \triangleq 1/4\sum_{i=1}^4 \mathbf{x}_{\text{est}_i}^T\mathbf{L}_i\mathbf{x}_{\text{est}_i}$.

The whole simulation is repeated 20 times. In all simulations, graphs and $\mathbf{x}_i$'s are kept the same, but different noise signals are added to $\mathbf{x}$. Average values over all simulations are denoted by $\overline{\text{SNR}}_{\text{input}}$ and $\overline{\text{Smoothness}}_{\text{av}}$. Figs. 3 and 4 show simulation results for different values of $\overline{\text{SNR}}_{\text{input}}$. As seen in Fig. 3, for very noisy inputs (*i.e.* smaller values of $\overline{\text{SNR}}_{\text{input}}$), the performance of (5) is better than (3), as was expected. In Fig. 4, the dashed line indicates $\text{Smoothness}_{\text{input}} \triangleq 1/4\sum_{i=1}^4 \mathbf{x}_i^T\mathbf{L}_i\mathbf{x}_i$. We observe that the $\overline{\text{Smoothness}}_{\text{av}}$ resulting from (5) is around the correct value of $\text{Smoothness}_{\text{input}}$, while (3), being forced to satisfy the exact equality $\mathbf{x} = \sum_{i=1}^K \mathbf{x}_i$, cannot find smooth enough signals.

F. Experiment 6: A proof of concept with real data

As a proof of concept, in this experiment, the performance of (7) is validated on a single-observation source separation problem that is generated by mixing two natural images with naively constructed graphs. Firstly, two 64 by 64 pixel gray-scale images are selected. They are scaled between 0 and 255 and reshaped to 4096-vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$. Then, for each image, a graph with 4096 vertices is created by connecting each pixel to its eight neighbors, where the weight between each pair of neighboring nodes is set equal to $w_{ijk} \triangleq 1/(|x_{ki}-x_{kj}|+0.001)$, $k = 1, 2$, in which x_{ki} is the i-th entry of the k-th graph signal ($\mathbf{x}_k, k = 1, 2$), and w_{ijk} is the weight between nodes (pixels) i and j of the k-th graph. With this choice of graph, the signals over graphs are relatively smooth signals, although not exactly band-limited (for example, Fig. 5 shows the first 500 frequency components of a 64 by 64 pixel image). Then, $\mathbf{x}$ is created as $\sum_{i=1}^2 \mathbf{x}_i$. The goal of this experiment is to separate two smooth graph signals, and verify whether or not they are close to the original underlying signals. Note that this is only a proof of concept, because firstly in practice we often do not have access to the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, and secondly although $\mathbf{x}_1$ and $\mathbf{x}_2$ are relatively smooth on these artificially constructed graphs, they are not exactly band-limited. Since $\mathbf{x}_1$ and $\mathbf{x}_2$ are not exactly sparse in the frequency domain, we do not use (10) for the decomposition, and use only (7). Moreover, as in Experiment 3, w_i in (7) is set equal to $(i-2)\times s+1$, with $s = 40$.

Table IV shows the results for two different sets of images. The SNR_i is defined as $\text{SNR}_i \triangleq 10\log_{10}\big(\|\mathbf{z}_i\|_2^2/\|\mathbf{z}_i - \mathbf{x}_{\text{est}_i}\|_2^2\big)$, $\forall i = 1, 2$, where $\mathbf{z}_i \triangleq \mathbf{x}_i - \bar{x}_i\mathbf{1}$. Moreover, Fig. 6 depicts the two sets of tested images, and their reconstruction. Note that to visualize the mixed image, $\mathbf{x}/2$ is actually plotted (to have pixel values between 0 and 255). We note that since the estimated signals have zero DC values, in order to bring back pixel values to the range 0 to 255, the constant 127 is

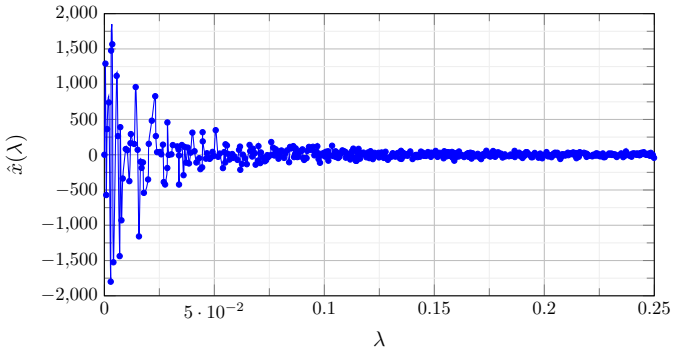


Fig. 5: First 500 frequency components of a 64 by 64 pixel image.

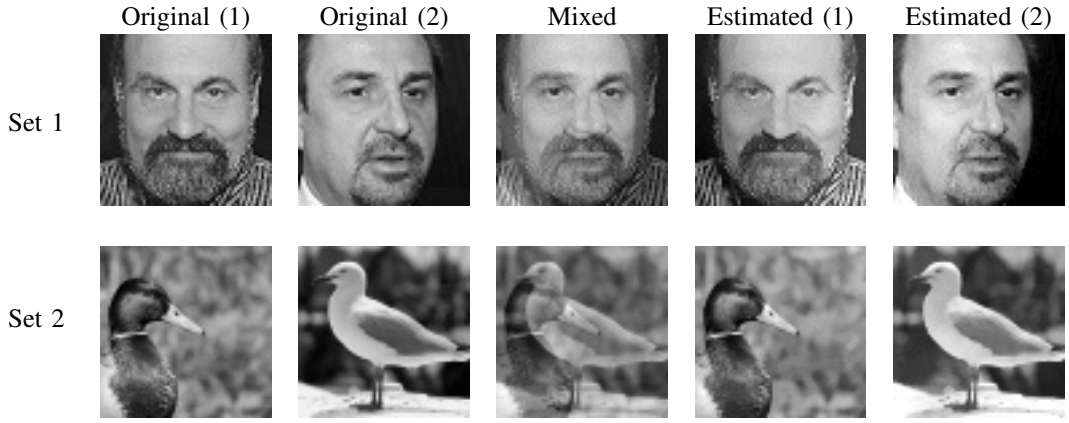


Fig. 6: Graph signal separation results for two different sets of images.

TABLE IV: Performance (in dB) of (7) for two different sets of images.

	SNR_1	SNR_2
Set 1	9.62	9.67
Set 2	9.07	13.87

added to them, and then the values above 255 and under 0 are cropped.

As it is seen both in Table IV and Fig. 6, the algorithm has been successful in recovering the two mixed images. It is observed that the quality of the recovery is satisfactory, despite the fact that the original signals are not exactly band-limited.

G. Experiment 7: Comparison with the method of [14]

In this experiment, the performance of (7) and the method of [14] are compared[2] in separating two mixed natural images (as in the previous experiment). Firstly, two 16×16 pixel gray-scale images[3] are selected, and graphs, graph signals $(\mathbf{x}_1, \mathbf{x}_2)$, and $\mathbf{x}$ are created similar to Experiment 6. Then, the images are separated once using (7), and once using the method of [14]. When using (7), the parameters are chosen as the previous experiment. In [14], the assumption is that $\mathbf{x}_i$'s are diffused graph signals passed through unknown linear graph filters $\mathbf{H}_i = \sum_{l=0}^{L-1} h_{il}\mathbf{S}_i^l$, and the degree of filters $(L-1)$ is known. To do the separation, we set $\mathbf{S}_i = \mathbf{W}_i$, and check the results for different values of L. The best value of SNRs which is for $L = 2$ is reported in Table V.

The resulting separated images of the two methods are scaled 4 times and shown in Fig 7. Note that to visualize the separated graph signals derived by the method of [14], the values above 255 and under 0 are cropped. Based on this figure and Table V, the performance of (7) is highly better than that of [14] in separating mixed images. This was actually expected because the method of [14] assumes a very specific model for the source graph signals, which does not hold here.

[2]The source code of the method of [14] has been taken from http://github.com/iglesias/gsp_bss.

[3]In this experiment, we had to use images smaller than the previous experiment because the method of [14] did not converge for the graphs as large as the previous experiment.

TABLE V: Performance (in dB) of two demixing methods.

	SNR_1	SNR_2
Solved by (7)	9.89	9.54
Solved by [14]	4.89	4.40

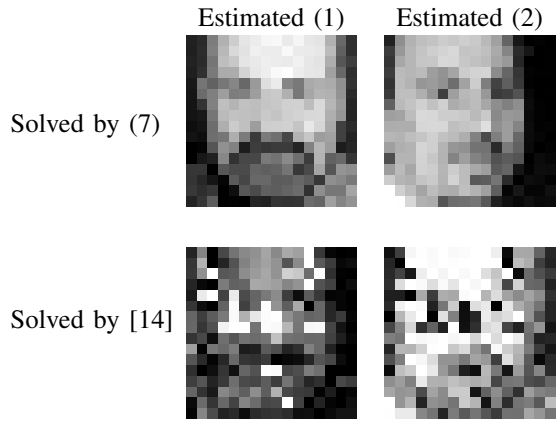


Fig. 7: Graph signal separation results for two demixing methods.

VI. Conclusion

In this paper, an approach was proposed to decompose a single signal into the summation of K smooth graph signals residing on different layers of a known multi-layer graph. It was shown that up to the indeterminacy of the DC values of signals, this approach provides a unique solution, both for noiseless and noisy signals. Moreover, for noiseless data, two modified methods were proposed to generalize the smoothness assumption to sparsity in the frequency domain; one for the case the frequency supports are a priori known, and one for the case they are not known. These generalizations can be extended to noisy data, in exactly similar manner. Simulation results confirmed on the effectiveness of these methods.

It is worth mentioning that in the BSS point of view, the recoverability of the sources remains ad-hoc in this paper (as in many original BSS papers, *e.g.* [26], [27]): it is true that the decomposition is unique, but can the output signals be rigorously proved to be exactly the original sources?

And under what conditions? Or if they are not exactly the original sources, how far are they from the original sources? In practice, if the original graph signals are known to be smooth, one may hope that they are not too far from the unique smoothest ones given by Theorems 1 through 3, and hence, one may hope that the sources estimated by the approaches of this paper are at least good approximations of the original sources. These questions need more investigations and would be a subject for future research.

APPENDIX A
PROOF OF THEOREM (1)

We firstly show that the solution of (2) is not generally unique. Then, we prove that different solutions differ only in their DC values.

A. Uniqueness

Since $\mathbf{x}$ is equal to the summation of K graph signals, $\mathbf{x}_K = \mathbf{x} - \sum_{i=1}^{K-1} \mathbf{x}_i$ holds. Therefore, (2) simplifies to

$$\underset{\{\mathbf{x}_i\}_{i=1}^{K-1}}{\text{minimize}} \quad \underbrace{\sum_{i=1}^{K-1} \mathbf{x}_i^T \mathbf{L}_i \mathbf{x}_i + (\mathbf{x} - \sum_{i=1}^{K-1} \mathbf{x}_i)^T \mathbf{L}_K (\mathbf{x} - \sum_{i=1}^{K-1} \mathbf{x}_i)}_{f(\mathbf{x}_1,\ldots,\mathbf{x}_{K-1})}. \tag{11}$$

Define $\tilde{\mathbf{x}} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_{K-1}^T]^T$. The quadratic problem (11) is convex, so by putting $\partial f / \partial \tilde{\mathbf{x}}$ equal to zero, $\{\mathbf{x}_i\}_{i=1}^K$ is computed.

$$\frac{\partial f}{\partial \tilde{\mathbf{x}}} = 2 \begin{bmatrix} (\mathbf{L}_1 + \mathbf{L}_K)\mathbf{x}_1 + \mathbf{L}_K \mathbf{x}_2 + \ldots + \mathbf{L}_K \mathbf{x}_{K-1} - \mathbf{L}_K \mathbf{x} \\ \mathbf{L}_K \mathbf{x}_1 + (\mathbf{L}_2 + \mathbf{L}_K)\mathbf{x}_2 + \ldots + \mathbf{L}_K \mathbf{x}_{K-1} - \mathbf{L}_K \mathbf{x} \\ \vdots \\ \mathbf{L}_K \mathbf{x}_1 + \ldots + (\mathbf{L}_{K-1} + \mathbf{L}_K)\mathbf{x}_{K-1} - \mathbf{L}_K \mathbf{x} \end{bmatrix},$$

$$\frac{\partial^2 f}{\partial \tilde{\mathbf{x}}^2} = 2 \begin{bmatrix} \mathbf{L}_1 + \mathbf{L}_K & \mathbf{L}_K & \ldots & \mathbf{L}_K \\ \mathbf{L}_K & \mathbf{L}_2 + \mathbf{L}_K & \ldots & \mathbf{L}_K \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}_K & \mathbf{L}_K & \ldots & \mathbf{L}_{K-1} + \mathbf{L}_K \end{bmatrix}. \tag{12}$$

Define $\mathbf{H} = \partial^2 f / \partial \tilde{\mathbf{x}}^2$. So, $\partial f / \partial \tilde{\mathbf{x}} = 0$ leads to

$$\mathbf{H}\tilde{\mathbf{x}} = 2 \begin{bmatrix} (\mathbf{L}_K \mathbf{x})^T & (\mathbf{L}_K \mathbf{x})^T & \ldots & (\mathbf{L}_K \mathbf{x})^T \end{bmatrix}^T. \tag{13}$$

Uniqueness of $\tilde{\mathbf{x}}$ depends on $\det(\mathbf{H})$. Since the optimization problem in (11) is convex, $\det(\mathbf{H}) \geq 0$, but not necessarily $\det(\mathbf{H}) > 0$. We show that actually $\det(\mathbf{H}) = 0$ by finding a leading principal submatrix of $\mathbf{H}$ with a zero determinant. For this purpose, define $\mathbf{\Gamma}$ as the $2N$-th order leading principal submatrix of $\mathbf{H}$, so

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{L}_1 + \mathbf{L}_K & \mathbf{L}_K \\ \mathbf{L}_K & \mathbf{L}_2 + \mathbf{L}_K \end{bmatrix}. \tag{14}$$

Since $\mathbf{v}_1$ (the all-one vector normalized to unit norm) is the common eigenvector of $\mathbf{L}_1$, $\mathbf{L}_2$ and $\mathbf{L}_K$ corresponding to zero eigenvalue,

$$\begin{bmatrix} \mathbf{L}_1 + \mathbf{L}_K & \mathbf{L}_K \\ \mathbf{L}_K & \mathbf{L}_2 + \mathbf{L}_K \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_1 \end{bmatrix} = \mathbf{0}, \tag{15}$$

where $\mathbf{0}$ stands for the all-zero vector. The latter implies that $\det(\mathbf{\Gamma}) = 0 \Rightarrow \det(\mathbf{H}) = 0$. Thus, $\mathbf{H}$ is singular, and problem (2) has several solutions.

B. The difference of solutions

Suppose both $\mathbf{x}_1, \ldots, \mathbf{x}_K$ and $\mathbf{x}'_1, \ldots, \mathbf{x}'_K$ are solutions of (2). So, based on (13),

$$\mathbf{L}_1 \mathbf{x}_1 = \mathbf{L}_2 \mathbf{x}_2 = \ldots = \mathbf{L}_{K-1} \mathbf{x}_{K-1} = \mathbf{L}_K (\mathbf{x} - \sum_{i=1}^{K-1} \mathbf{x}_i), \tag{16}$$

$$\mathbf{L}_1 \mathbf{x}'_1 = \mathbf{L}_2 \mathbf{x}'_2 = \ldots = \mathbf{L}_{K-1} \mathbf{x}'_{K-1} = \mathbf{L}_K (\mathbf{x} - \sum_{i=1}^{K-1} \mathbf{x}'_i). \tag{17}$$

Define $\mathbf{e}_i \triangleq \mathbf{x}_i - \mathbf{x}'_i, \ \forall i = 1, \ldots, K$. Therefore, subtracting (17) form (16) results in

$$\mathbf{L}_1 \mathbf{e}_1 = \mathbf{L}_2 \mathbf{e}_2 = \ldots = \mathbf{L}_{K-1} \mathbf{e}_{K-1} = \mathbf{L}_K (\mathbf{0} - \sum_{i=1}^{K-1} \mathbf{e}_i). \tag{18}$$

Since (16) is the necessary and sufficient condition for $\mathbf{x}_1, \ldots, \mathbf{x}_K$ to be a solution of (2) for an arbitrary $\mathbf{x}$, (18) shows that $\mathbf{e}_1, \ldots, \mathbf{e}_K$ is a solution of (2) for decomposing an all-zero vector. Hence, $\mathbf{e}_1, \ldots, \mathbf{e}_K$ are the solution of the optimization problem

$$\underset{\{\mathbf{e}_i\}_{i=1}^K}{\text{minimize}} \quad \sum_{i=1}^{K} \mathbf{e}_i^T \mathbf{L}_i \mathbf{e}_i \quad \text{s.t.} \quad \sum_{i=1}^{K} \mathbf{e}_i = \mathbf{0}. \tag{19}$$

The cost function of the above optimization problem is always non-negative, and $\mathbf{e}'_i \triangleq \mathbf{0}, \forall i$ makes it zero and satisfies the constraint as well. So, $\mathbf{e}'_i = \mathbf{0}, \forall i$ is a solution of the above problem. Since the cost function of the above problem cannot be negative, every other solution of (19) should make the cost function zero, too. Since $\mathbf{e}_i^T \mathbf{L}_i \mathbf{e}_i \geq 0, \ \forall i = 1, \ldots, K$, every solution of (19) should satisfy

$$\begin{cases} \mathbf{e}_i^T \mathbf{L}_i \mathbf{e}_i = 0 & \forall i = 1, \ldots, K \\ \sum_{i=1}^K \mathbf{e}_i = \mathbf{0} \end{cases}. \tag{20}$$

Substituting the eigenvalue decomposition of $\mathbf{L}_i$ into $\mathbf{e}_i^T \mathbf{L}_i \mathbf{e}_i = 0$ leads to

$$\mathbf{e}_i^T \mathbf{V}_i \mathbf{\Lambda}_i \mathbf{V}_i^T \mathbf{e}_i = \mathbf{y}_i^T \mathbf{\Lambda}_i \mathbf{y}_i = \sum_{j=1}^{N} \lambda_{ij} y_{ij}^2 = 0, \tag{21}$$

where $\mathbf{y}_i = \mathbf{V}_i^T \mathbf{e}_i$. Since all the graphs are connected, only $\lambda_{i1} = 0$. So, (21) holds, if

$$y_{ij} = \begin{cases} 0 & \forall j = 2, \ldots, N \\ \alpha_i & j = 1 \end{cases}, \tag{22}$$

where α_i is an arbitrary value. Noting that the eigenvector corresponding to λ_{i1} is proportional to an all-one vector, $\mathbf{e}_i$ is found as

$$\mathbf{e}_i = \mathbf{V}_i \mathbf{y}_i = \sum_{j=1}^{N} \mathbf{v}_{ij} y_{ij} = \alpha_i \mathbf{1}, \tag{23}$$

where $\mathbf{1}$ is an all-one vector, and $\mathbf{V}_i = [\mathbf{v}_{i1}, \ldots, \mathbf{v}_{iN}]$. So,

$$\mathbf{x}_i = \mathbf{x}'_i + \alpha_i \mathbf{1}, \quad \forall i = 1, \ldots, K, \tag{24}$$

which means that different solutions of (2) are only different in their DC values. ■

APPENDIX B
PROOF OF THEOREM (2)

It is firstly shown that the solution of (4) is not generally unique, and, then, it is proved that different solutions differ only in their DC values.

A. Uniqueness

To find $\{\mathbf{x}_i\}_{i=1}^K$, $\partial g/\partial \check{\mathbf{x}}$ should be set equal to zero, where $g(\mathbf{x}_1,\ldots,\mathbf{x}_K)$ is the objective function of (4), and $\check{\mathbf{x}} = [\mathbf{x}_1^T,\ldots,\mathbf{x}_K^T]^T$.

$$\begin{aligned}\frac{\partial g}{\partial \check{\mathbf{x}}} &= 2\begin{bmatrix}(\mathbf{I}+\gamma_1\mathbf{L}_1)\mathbf{x}_1+\mathbf{x}_2+\ldots+\mathbf{x}_K-\mathbf{x}\\ \mathbf{x}_1+(\mathbf{I}+\gamma_2\mathbf{L}_2)\mathbf{x}_2+\ldots+\mathbf{x}_K-\mathbf{x}\\ \vdots\\ \mathbf{x}_1+\mathbf{x}_2+\ldots+(\mathbf{I}+\gamma_K\mathbf{L}_K)\mathbf{x}_K-\mathbf{x}\end{bmatrix},\\ \frac{\partial^2 g}{\partial \check{\mathbf{x}}^2} &= 2\begin{bmatrix}\mathbf{I}+\gamma_1\mathbf{L}_1 & \mathbf{I} & \ldots & \mathbf{I}\\ \mathbf{I} & \mathbf{I}+\gamma_2\mathbf{L}_2 & \ldots & \mathbf{I}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{I} & \mathbf{I} & \ldots & \mathbf{I}+\gamma_K\mathbf{L}_K\end{bmatrix}.\end{aligned} \tag{25}$$

Define, $\mathbf{H} = \partial^2 g/\partial \check{\mathbf{x}}^2$. So, $\partial g/\partial \check{\mathbf{x}} = 0$ leads to

$$\mathbf{H}\check{\mathbf{x}} = 2\begin{bmatrix}\mathbf{x}^T & \mathbf{x}^T & \ldots & \mathbf{x}^T\end{bmatrix}^T. \tag{26}$$

Similar to Appendix A-A, define $\mathbf{\Delta}$ as the $2N$-th order leading principal submatrix of $\mathbf{H}$ as follows, to show that there exists a leading principal submatrix of $\mathbf{H}$ with zero determinant:

$$\mathbf{\Delta} = \begin{bmatrix}\mathbf{I}+\gamma_1\mathbf{L}_1 & \mathbf{I}\\ \mathbf{I} & \mathbf{I}+\gamma_2\mathbf{L}_2\end{bmatrix}. \tag{27}$$

Since $\gamma_1\mathbf{L}_1, \gamma_2\mathbf{L}_2 \geq 0$, $(\mathbf{I}+\gamma_1\mathbf{L}_1)$ and $(\mathbf{I}+\gamma_2\mathbf{L}_2)$ are invertible. Based on Equation (0.8.5.1) of [28] for the determinant of 2-by-2 block matrices, we have that

$$\det(\mathbf{\Delta}) = \det(\mathbf{I}+\gamma_1\mathbf{L}_1)\det((\mathbf{I}+\gamma_2\mathbf{L}_2)-(\mathbf{I}+\gamma_1\mathbf{L}_1)^{-1}). \tag{28}$$

The smallest eigenvalue of $(\mathbf{I}+\gamma_1\mathbf{L}_1)$ and $(\mathbf{I}+\gamma_2\mathbf{L}_2)$ are equal to one and both correspond to the constant eigenvector, $\mathbf{v}_1$. Since $(\mathbf{I}+\gamma_1\mathbf{L}_1)$ is invertible, its eigenvalues are the inverses of the eigenvalues of $(\mathbf{I}+\gamma_1\mathbf{L}_1)^{-1}$ with the same eigenvectors. Therefore, the largest eigenvalue of $(\mathbf{I}+\gamma_1\mathbf{L}_1)^{-1}$ is equal to one, and the following equality holds:

$$((\mathbf{I}+\gamma_2\mathbf{L}_2)-(\mathbf{I}+\gamma_1\mathbf{L}_1)^{-1})\mathbf{v}_1 = \mathbf{0}. \tag{29}$$

Hence, $\det((\mathbf{I}+\gamma_2\mathbf{L}_2)-(\mathbf{I}+\gamma_1\mathbf{L}_1)^{-1}) = 0 \Rightarrow \det(\mathbf{\Delta}) = 0 \Rightarrow \det(\mathbf{H}) = 0$. Thus, $\mathbf{H}$ is singular, and the solution of (4) is not unique.

B. The difference of solutions

Suppose both $\mathbf{x}_1,\ldots,\mathbf{x}_K$ and $\mathbf{x}'_1,\ldots,\mathbf{x}'_K$ are solutions of (4). Hence, based on (26),

$$\gamma_1\mathbf{L}_1\mathbf{x}_1 = \gamma_2\mathbf{L}_2\mathbf{x}_2 = \ldots = \gamma_K\mathbf{L}_K\mathbf{x}_K = \mathbf{x} - \sum_{i=1}^K \mathbf{x}_i, \tag{30}$$

$$\gamma_1\mathbf{L}_1\mathbf{x}'_1 = \gamma_2\mathbf{L}_2\mathbf{x}'_2 = \ldots = \gamma_K\mathbf{L}_K\mathbf{x}'_K = \mathbf{x} - \sum_{i=1}^K \mathbf{x}'_i. \tag{31}$$

Define $\mathbf{e}_i \triangleq \mathbf{x}_i - \mathbf{x}'_i$, $\forall i = 1,\ldots,K$. Therefore, subtracting (31) form (30) leads to

$$\gamma_1\mathbf{L}_1\mathbf{e}_1 = \gamma_2\mathbf{L}_2\mathbf{e}_2 = \ldots = \gamma_K\mathbf{L}_K\mathbf{e}_K = \mathbf{0} - \sum_{i=1}^{K-1}\mathbf{e}_i, \tag{32}$$

which means that similar to (30), an all-zero vector should be decomposed into the summation of $\mathbf{e}_1,\ldots,\mathbf{e}_K$. Hence, $\mathbf{e}_i$, $\forall i = 1,\ldots,K$ can be calculated through the quadratic problem

$$\underset{\{\mathbf{e}_i\}_{i=1}^K}{\text{minimize}} \quad \|\sum_{i=1}^K \mathbf{e}_i\|_2^2 + \sum_{i=1}^K \gamma_i \mathbf{e}_i^T\mathbf{L}_i\mathbf{e}_i\ . \tag{33}$$

The cost function of the above optimization problem is always non-negative, and vanishes for $\mathbf{e}'_i \triangleq \mathbf{0}, \forall i$. Hence, $\mathbf{e}'_i = \mathbf{0}, \forall i$ is a solution of (33), and taking into account the convexity of the problem, every other local solution of it should make the cost function zero as well. Therefore, noting that the individual term of the above cost function are all non-negative, we have

$$\begin{cases}\mathbf{e}_i^T\mathbf{L}_i\mathbf{e}_i = 0 & \forall i = 1,\ldots,K\\ \sum_{i=1}^K \mathbf{e}_i = \mathbf{0}\end{cases}, \tag{34}$$

which is similar to (20). The rest of the proof is like Appendix A-B, and so different solutions of (4) are different only in their DC values. ■

APPENDIX C
PROOF OF THEOREM (3)

Since $\hat{x}_{i1} = 0, \forall i = 1,\ldots,K$, (7) simplifies to

$$\underset{\{\hat{\mathbf{x}}'_i\}_{i=1}^K}{\text{minimize}} \quad \sum_{j=2}^N w_j \sum_{i=1}^K \hat{x}_{ij}^2 \quad \text{s.t.} \quad \mathbf{z} = \sum_{i=1}^K \mathbf{V}'_i\hat{\mathbf{x}}'_i\ , \tag{35}$$

where $\mathbf{V}'_i = [\mathbf{v}_{i2},\ldots,\mathbf{v}_{iN}]$ and $\hat{\mathbf{x}}'_i = [\hat{x}_{i2},\ldots,\hat{x}_{iN}]^T$. Consider the objective function of (35) as $f(\hat{\mathbf{x}}'_1,\ldots,\hat{\mathbf{x}}'_K)$, and define $\hat{\mathbf{x}}' = [(\hat{\mathbf{x}}'_1)^T,\ldots,(\hat{\mathbf{x}}'_K)^T]^T$, then

$$\mathbf{H} = \frac{\partial^2 f}{\partial \hat{\mathbf{x}}'^2} = 2\begin{bmatrix}\text{diag}(\mathbf{w}) & \mathbf{0} & \ldots & \mathbf{0}\\ \mathbf{0} & \text{diag}(\mathbf{w}) & \ldots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \ldots & \text{diag}(\mathbf{w})\end{bmatrix}, \tag{36}$$

where $\mathbf{w} = [w_2,\ldots,w_N]^T$, and diag(.) returns a diagonal matrix. Since $w_i > 0, \forall i = 2,\ldots,N$, for $m/2 < \text{argmin}(\mathbf{w})$, $\mathbf{H} \geq m\mathbf{I}$, which means that f is strongly convex. Based on [29], as f is a continuous and strongly convex function which is minimized over a closed convex set, it has a unique minimizer. ■

REFERENCES

[1] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[2] Aliaksei Sandryhaila and José M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[3] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin, "The structure and dynamics of multilayer networks," *Physics reports*, vol. 544, no. 1, pp. 1–122, 2014.

[4] Barry Bentley, Robyn Branicky, Christopher L Barnes, Yee Lian Chew, Eviatar Yemini, Edward T Bullmore, Petra E Vértes, and William R Schafer, "The multilayer connectome of caenorhabditis elegans," *PLoS computational biology*, vol. 12, no. 12, pp. e1005283, 2016.

[5] Manlio De Domenico, "Multilayer modeling and analysis of human brain networks," *Giga Science*, vol. 6, no. 5, pp. gix004, 2017.

[6] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter, "Multilayer networks," *Journal of Complex Networks*, vol. 3, no. 2, pp. 203–271, September 2014.

[7] Pierre Comon and Christian Jutten, *Handbook of Blind Source Separation, Independent Component Analysis and Applications*, Academic Press (Elsevier), Feb. 2010.

[8] Florian Blöchl, Andreas Kowarsch, and Fabian J. Theis, "Second-order source separation based on prior knowledge realized in a graph model," in *Latent Variable Analysis and Signal Separation*, Vincent Vigneron, Vicente Zarzoso, Eric Moreau, Rémi Gribonval, and Emmanuel Vincent, Eds., Berlin, Heidelberg, 2010, pp. 434–441, Springer Berlin Heidelberg.

[9] Andreas Kowarsch, Florian Blöchl, Sebastian Bohl, Maria Saile, Norbert Gretz, Ursula Klingmüller, and Fabian J. Theis, "Knowledge-based matrix factorization temporally resolves the cellular responses to il-6 stimulation," *BMC Bioinformatics*, vol. 11, 2010.

[10] Jari Miettinen, Eyal Nitzan, Sergiy A. Vorobyov, and Esa Ollila, "Graph signal processing meets blind source separation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2585–2599, 2021.

[11] Jari Miettinen, Sergiy A. Vorobyov, and Esa Ollila, "Blind source separation of graph signals," in *2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020 - Proceedings*, United States, May 2020, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 5645–5649, IEEE, IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP ; Conference date: 04-05-2020 Through 08-05-2020.

[12] Aref Einizade, Sepideh Hajipour Sardouie, and Mohammad B. Shamsollahi, "Simultaneous graph learning and blind separation of graph signal sources," *IEEE Signal Processing Letters*, vol. 28, pp. 1495–1499, 2021.

[13] Aref Einizade and Sepideh Hajipour Sardouie, "Robust blind separation of smooth graph signals using minimization of graph regularized mutual information," *Digital Signal Processing*, vol. 132, pp. 103792, 2022.

[14] Fernando J. Iglesias, Santiago Segarra, Samuel Rey-Escudero, Antonio G. Marques, and David Ramírez, "Demixing and blind deconvolution of graph-diffused sparse signals," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4189–4193.

[15] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[16] Yonina C. Eldar and Moshe Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.

[17] Michael Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer Publishing Company, Incorporated, 1st edition, 2010.

[18] Nathanaël Perraudin, Johan Paratte, David I. Shuman, Lionel Martin, Vassilis Kalofolias, Pierre Vandergheynst, and David K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014.

[19] Steven Diamond and Stephen Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[20] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[21] Nitaigour P. Mahalik, *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*, Springer-Verlag, Berlin, Heidelberg, 2006.

[22] Albert-László Barabási and Réka Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[23] Jeong Han Kim and Van H. Vu, "Generating random regular graphs," in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 2003, STOC '03, p. 213–222, Association for Computing Machinery.

[24] Michelle Girvan and Mark E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[25] Paul Erdös and Alfréd Rényi, "On random graphs I," *Publicationes Mathematicate*, vol. 6, pp. 290–297, 1959.

[26] Jeanny Hérault and Christian Jutten, "Space or time adaptive signal processing by neural networks models," in *Intern. Conf. on Neural Networks for Computing*, Snowbird (Utah, USA), 1986, pp. 206–211.

[27] David N. Levin, "Performing nonlinear blind source separation with signal invariants," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2131–2140, 2010.

[28] Roger A. Horn and Charles R. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 2nd edition, 2013.

[29] "Existence of minimizer for strongly convex function on closed, convex set," Mathematics Stack Exchange, URL: https://math.stackexchange.com/q/4477709 (Accessed 2022-06-22).