Introducing Hypergraph Signal Processing: Theoretical Foundation and Practical Applications

Songyang Zhang, Zhi Ding, *Fellow, IEEE*, and Shuguang Cui, *Fellow, IEEE*

***Abstract*—Signal processing over graphs has recently attracted significant attention for dealing with the structured data. Normal graphs, however, only model pairwise relationships between nodes and are not effective in representing and capturing some high-order relationships of data samples, which are common in many applications, such as Internet of Things (IoT). In this article, we propose a new framework of hypergraph signal processing (HGSP) based on the tensor representation to generalize the traditional graph signal processing (GSP) to tackle high-order interactions. We introduce the core concepts of HGSP and define the hypergraph Fourier space. We then study the spectrum properties of hypergraph Fourier transform (HGFT) and explain its connection to mainstream digital signal processing. We derive the novel hypergraph sampling theory and present the fundamentals of hypergraph filter design based on the tensor framework. We present HGSP-based methods for several signal processing and data analysis applications. Our experimental results demonstrate significant performance improvement using our HGSP framework over some traditional signal processing solutions.**

***Index Terms*—Data analysis, hypergraph, signal processing, tensor.**

I. Introduction

GRAPH-theoretic tools have recently found broad applications in data science owing to their power to model complex relationships in the large structured data sets [1]. Big data, such as those representing social network interactions, Internet of Things (IoT) intelligence, biological connections, mobility, and traffic patterns, often exhibit complex structures that are challenging to many traditional tools [2]. Thankfully, graphs provide good models for many such data sets as well as the underlying complex relationships. A data set with N data points can be modeled as a graph of N vertices, whose internal relationships can be captured by edges. For example, subscribing users in a communication or social network can be modeled as nodes while the physical interconnections or social relationships among users are represented as edges [3].

Taking an advantage of graph models in characterizing the complex data structures, graph signal processing (GSP) has emerged as an exciting and promising new tool for processing the large data sets with complex structures. A typical application of GSP is in image processing, where image pixels are modeled as graph signals embedding in nodes while pairwise similarities between pixels are captured by edges [6]. By modeling the images using graphs, tasks, such as image segmentation can take advantage of graph partition and GSP filters. Another example of GSP applications is in processing data from the sensor networks [5]. Based on the graph models directly built over the network structures, a graph Fourier space could be defined according to the eigenspace of a representing graph matrix, such as the Laplacian or adjacency matrix, to facilitate data processing operations, such as denoising [7], filter banks [8], and compression [9].

Despite many demonstrated successes, the GSP defined over normal graphs also exhibits certain limitations. First, normal graphs cannot capture high-dimensional interactions describing multilateral relationships among multiple nodes, which are critical for many practical applications. Since each edge in a normal graph only models the pairwise interactions between the two nodes, the traditional GSP can only deal with the pairwise relationships defined by such edges. In reality, however, complex relationships may exist among a cluster of nodes, for which the use of pairwise links between every two nodes cannot capture their multilateral interactions [16]. In biology, for example, a trait may be attributed to multiple interactive genes [17] shown in Fig. 1(a), such that a quadrilateral interaction is more informative and powerful here. Another example is the social network with online social communities called folksonomies, where trilateral interactions occur among users, resources, and annotations [15], [40]. Second, a normal graph can only capture a typical single-tier relationship with the matrix representation. In complex systems and data sets, however, each node may have several traits such that there exist multiple tiers of interactions between two nodes. In a cyber-physical system, for example, each node usually contains two components, i.e., the physical component and the cyber component, for which there exist two tiers of connections between a pair of nodes. Generally, such multitier relationships can be modeled as multilayer networks, where each layer represents one tier of interactions [11]. However, normal graphs cannot model the interlayer interactions simply, and the corresponding matrix representations are unable

Manuscript received August 12, 2019; revised October 2, 2019; accepted October 21, 2019. Date of publication October 30, 2019; date of current version January 10, 2020. This work was supported in part by NSF under Grant DMS-1622433 and Grant CNS-1824553. *(Corresponding author: Shuguang Cui.)*

S. Zhang and Z. Ding are with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA 95616 USA (e-mail: sydzhang@ucdavis.edu; zding@ucdavis.edu).

S. Cui was with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA 95616 USA. He is now with the Shenzhen Research Institute of Big Data and Future Network of Intelligence Institute (FNii), the Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: robert.cui@gmail.com).

Digital Object Identifier 10.1109/JIOT.2019.2950213

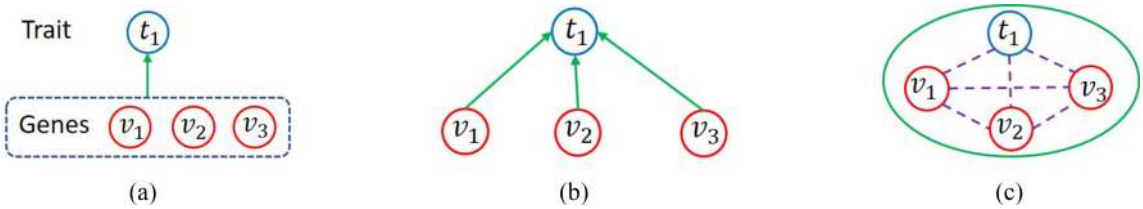

Fig. 1. Example of multilateral relationships. (a) Example of a trait and genes: the trait t_1 is triggered by three genes v_1, v_2, v_3, and the genes may also influence each other. (b) Example of pairwise relationships: arrow links represent the influences from genes, whereas the potential interactions among genes cannot be captured. (c) Example of multilateral relationships: the solid circular line represents a quadrilateral relationship among four nodes, while the purple dashed lines represent the potential internode interactions in this quadrilateral relationship.

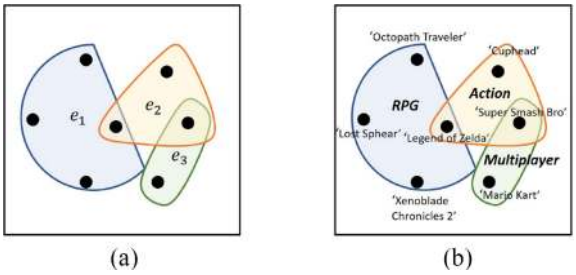

Fig. 2. Hypergraphs and applications. (a) Example of hypergraphs: the hyperedges are the overlaps covering nodes with different colors. (b) Game data set modeled by hypergraphs: each node is a specific game and each hyperedge is a category of games.

to distinguish different tiers of relationships efficiently since they describe entries for all layers equivalently [10], [14]. Thus, the traditional GSP based on the matrix analysis has far been unable to efficiently handle such complex relationships. Clearly, there is a need for a more general graph model and GSP concept to remedy the aforementioned shortcomings faced with the traditional GSP.

To find a more general model for complex data structures, we venture into the area of high-dimensional graphs known as hypergraphs. The hypergraph theory is increasingly playing an important role in graph theory and data analysis, especially for analyzing high-dimensional data structures and interactions [18]. A hypergraph consists of nodes and hyperedges connecting more than two nodes [19]. As an example, Fig. 2(a) shows a hypergraph example with three hyperedges and seven nodes, whereas Fig. 2(b) provides a corresponding data set modeled by this hypergraph. Indeed, a normal graph is a special case of a hypergraph, where each hyperedge degrades to a simple edge that only involves exactly two nodes.

Hypergraphs have found successes by generalizing the normal graphs in many applications, such as clustering [39], classification [22], and prediction [23]. Moreover, a hypergraph is an alternative representation for a multilayer network and is useful when dealing with the multitier relationships [12], [13]. Thus, a hypergraph is a natural extension of a normal graph in modeling signals of high-degree interactions. Presently, however, the literature provides a little coverage on hypergraph signal processing (HGSP). The only known work [4] proposed an HGSP framework based on a special hypergraph called complexes. In this article [4], hypergraph signals are associated with each hyperedge, but its framework is limited to cell complexes, which cannot suitably model many real-world data sets and applications. Another shortcoming of the framework in [4] is the lack of detailed analysis and application examples to demonstrate its practicability. In addition, the attempt in [4] to extend some key concepts from the traditional GSP simply fails due to the difference in the basic setups between the graph signals and hypergraph signals. In this article, we seek to establish a more general and practical HGSP framework, capable of handling arbitrary hypergraphs and naturally extending the traditional GSP concepts to handle the high-dimensional interactions. We will also provide a real application examples to validate the effectiveness of the proposed framework.

Compared with the traditional GSP, a generalized HGSP faces several technical challenges. The first problem lies in the mathematical representation of hypergraphs. Developing an algebraic representation of a hypergraph is the foundation of HGSP. Currently, there are two major approaches: matrix-based [32] and tensor-based [20]. The matrix-based method makes it hard to implement the hypergraph signal shifting while the tensor-based method is difficult to be understood conceptually. Another challenge is in defining signal shifting over the hyperedge. Signal shifting is easy to be defined as propagation along the link direction of a simple edge connecting the two nodes in a regular graph. However, each hyperedge in hypergraphs involves more than two nodes. How to model signal interactions over a hyperedge requires careful considerations. Other challenges include the definition and interpretation of hypergraph frequency.

To address the aforementioned challenges and generalize the traditional GSP into a more general hypergraph tool to capture high-dimensional interactions, we propose a novel tensor-based HGSP framework in this article. The main contributions in this article can be summarized as follows. Representing hypergraphs as tensors, we define a specific form of hypergraph signals and hypergraph signal shifting. We then provide an alternative definition of hypergraph Fourier space based on the orthogonal CANDECOMP/PARAFAC (CP) tensor decomposition, together with the corresponding hypergraph Fourier transform (HGFT). To better interpret the hypergraph Fourier space, we analyze the resulting hypergraph frequency properties, including the concepts of frequency and bandlimited signals. Analogous to the traditional sampling theory, we derive the conditions and properties for perfect signal recovery from samples in HGSP. We also provide the theoretical foundation for the HGSP filter designs. Beyond these, we provide several application examples of the proposed HGSP framework.

1) We introduce a signal compression method based on the new sampling theory to show the effectiveness of HGSP in describing the structured signals.

2) We apply HGSP in spectral clustering to show how the HGSP spectrum space acts as a suitable spectrum for hypergraphs.
3) We introduce an HGSP method for binary classification problems to demonstrate the practical application of HGSP in data analysis.
4) We introduce a filtering approach for the denoising problem to further showcase the power of HGSP.
5) Finally, we suggest several potential applicable backgrounds for HGSP, including IoT, social network, and natural language processing.

We compare the performance of HGSP-based methods with the traditional GSP-based methods and learning algorithms in all the above applications. All the features of HGSP make it an essential tool for IoT applications in the future.

We organize the rest of this article as follows. Section II first summarizes the preliminaries of the traditional GSP, tensors, and hypergraphs. In Section III, we then introduce the core definitions of HGSP, including the hypergraph signal, the signal shifting, and the hypergraph Fourier space, followed by the frequency interpretation and description of existing works in Section IV. We present some useful HGSP-based results, such as the sampling theory and filter design in Section V. With the proposed HGSP framework, we provide several potential applications of HGSP and demonstrate its effectiveness in Section VI, before presenting the final conclusions in Section VII.

II. Preliminaries

A. Overview of Graph Signal Processing

GSP is a recent tool used to analyze signals according to the graph models. Here, we briefly review the key relevant concepts of the traditional GSP [1], [2].

A data set with N data points can be modeled as a normal graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ consisting of a set of N nodes $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ and a set of edges $\mathcal{E}$. Each node of the graph $\mathcal{G}$ is a data point, whereas the edges describe the pairwise interactions between nodes. A graph signal represents the data associated with a node. For a graph with N nodes, there are N graph signals, which are defined as a signal vector $\mathbf{s} = [s_1 \quad s_2 \quad \cdots \quad s_N]^T \in \mathbb{R}^N$.

Usually, such a graph could be either described by an adjacency matrix $\mathbf{A_M} \in \mathbb{R}^{N\times N}$ where each entry indicates a pairwise link (or an edge), or by a Laplacian matrix $\mathbf{L_M} = \mathbf{D_M} - \mathbf{A_M}$ where $\mathbf{D_M} \in \mathbb{R}^{N\times N}$ is the diagonal matrix of degrees. Both the Laplacian matrix and the adjacency matrix can fully represent the graph structure. For convenience, we use a general matrix $\mathbf{F_M} \in \mathbb{R}^{N\times N}$ to represent either of them. Note that since the adjacency matrix is eligible in both directed and undirected graphs, it is more common in the GSP literature. Thus, the generalized GSP is based on the adjacency matrix [2] and the representing matrix refers to the adjacency matrix in this article unless specified otherwise.

With the graph representation $\mathbf{F_M}$ and the signal vector $\mathbf{s}$, the graph shifting is defined as

$$\mathbf{s}' = \mathbf{F_M}\mathbf{s}. \tag{1}$$

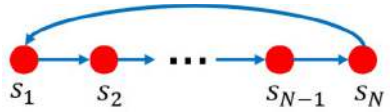

Fig. 3. Signal shifting over cyclic graph.

Here, the matrix $\mathbf{F_M}$ could be interpreted as a graph filter whose functionality is to shift the signals along link directions. Taking the cyclic graph shown in Fig. 3 as an example, its adjacency matrix is a shifting matrix

$$\mathbf{F_M} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \tag{2}$$

Typically, the shifted signal over the cyclic graph is calculated as $\mathbf{s}' = \mathbf{F_M}\mathbf{s} = [s_N \quad s_1 \quad \cdots \quad s_{N-1}]^T$, which shifts the signal at each node to its next node.

The graph spectrum space, also called the graph Fourier space, is defined based on the eigenspace of $\mathbf{F_M}$. Assume that the eigen-decomposition of $\mathbf{F_M}$ is

$$\mathbf{F_M} = \mathbf{V_M^{-1}}\Lambda\mathbf{V_M}. \tag{3}$$

The frequency components are defined by the eigenvectors of $\mathbf{F_M}$ and the frequencies are defined with respect to the eigenvalues. The corresponding graph Fourier transform is defined as

$$\hat{\mathbf{s}} = \mathbf{V_M}\mathbf{s}. \tag{4}$$

With the definition of the graph Fourier space, the traditional signal processing and learning tasks, such as denoising [33] and classification [73], could be solved within the GSP framework. More details about the specific topics of GSP, such as the frequency analysis, filter design, and spectrum representation have been discussed in [5], [52], and [84].

B. Introduction of Hypergraph

We begin with the definition of hypergraph and its possible representations.

Definition 1 (Hypergraph): A general hypergraph $\mathcal{H}$ is a pair $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ is a set of elements called vertices and $\mathcal{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$ is a set of nonempty multielement subsets of $\mathcal{V}$ called hyperedges. Let $M = \max\{|\mathbf{e}_i| : \mathbf{e}_i \in \mathcal{E}\}$ be the maximum cardinality of hyperedges, shorted as $m.c.e(\mathcal{H})$ of $\mathcal{H}$.

In a general hypergraph $\mathcal{H}$, different hyperedges may contain different numbers of nodes. The $m.c.e(\mathcal{H})$ denotes the number of vertices in the largest hyperedge. An example of a hypergraph with seven nodes, three hyperedges, and $m.c.e = 3$ is shown in Fig. 4.

From the definition, we see that a normal graph is a special case of a hypergraph if $M = 2$. The hypergraph is a natural extension of the normal graph to represent the high-dimensional interactions. To represent a hypergraph mathematically, there are two major methods based on the matrix and tensor, respectively. In the matrix-based method, a hypergraph

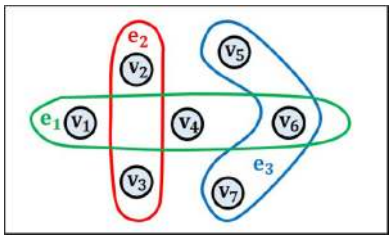

Fig. 4. Hypergraph $\mathcal{H}$ with seven nodes, three hyperedges, and $m.c.e(\mathcal{H}) = 3$, where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_7\}$ and $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. Three hyperedges are $\mathbf{e}_1 = \{\mathbf{v}_1, \mathbf{v}_4, \mathbf{v}_6\}$, $\mathbf{e}_2 = \{\mathbf{v}_2, \mathbf{v}_3\}$, and $\mathbf{e}_3 = \{\mathbf{v}_5, \mathbf{v}_6, \mathbf{v}_7\}$.

is represented by a matrix $\mathbf{G} \in \mathbb{R}^{N\times E}$, where E equals the number of hyperedges. The rows of the matrix represent the nodes, and the columns represent the hyperedges [19]. Thus, each element in the matrix indicates whether the corresponding node is involved in the particular hyperedge. Although such a matrix-based representation is simple in formation, it is hard to define and implement the signal processing directly as in GSP by using the matrix $\mathbf{G}$. Unlike the matrix-based method, tensor has better flexibility in describing the structures of the high-dimensional graphs [42]. More specifically, tensor can be viewed as an extension of matrix into high-dimensional domains. The adjacency tensor, which indicates whether nodes are connected, is a natural hypergraph counterpart to the adjacency matrix in the normal graph theory [51]. Thus, we prefer to represent the hypergraphs using tensors. In Section III-A, we will provide more details on how to represent the hypergraphs and signals in tensor forms.

C. Tensor Basics

Before we introduce our tensor-based HGSP framework, let us introduce some tensor basics to be used later. Tensors can effectively represent high-dimensional graphs [14]. Generally speaking, tensors can be interpreted as multidimensional arrays. The order of a tensor is the number of indices needed to label a component of that array [24]. For example, a third-order tensor has three indices. In fact, scalars, vectors, and matrices are all special cases of tensors: a scalar is a zeroth-order tensor; a vector is a first-order tensor; a matrix is a second-order tensor; and an M-dimensional array is an Mth-order tensor [10]. Generalizing a 2-D matrix, we represent the entry at the position $(i_1, i_2, \ldots, i_M)$ of an Mth-order tensor $\mathbf{T} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_M}$ by $t_{i_1 i_2\cdots i_M}$ in the rest of this article.

Below are some useful definitions and operations of tensor related to the proposed HGSP framework.

1) Symmetric and Diagonal Tensors:

1) A tensor is *super-symmetric* if its entries are invariant under any permutation of their indices [44]. For example, a third-order $\mathbf{T} \in \mathbb{R}^{I\times I\times I}$ is super-symmetric if its entries t_{ijk}'s satisfy

$$t_{ijk} = t_{jik} = t_{kij} = t_{kji} = t_{jik} = t_{jki} \quad i, j, k = 1, \ldots, I. \tag{5}$$

Analysis of super-symmetric tensors, which is shown to be bijectively related to homogeneous polynomials, could be found in [45] and [46].

2) A tensor $\mathbf{T} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is *super-diagonal* if its entries $t_{i_1 i_2\cdots i_N} \neq 0$ only if $i_1 = i_2 = \cdots = i_N$. For example, a third-order $\mathbf{T} \in \mathbb{R}^{I\times I\times I}$ is super-diagonal if its entries $t_{iii} \neq 0$ for $i = 1, 2, \ldots, I$, while all other entries are zero.

2) Tensor Operations: Tensor analysis is developed based on tensor operations. Some tensor operations are commonly used in our HGSP framework [48]–[50].

1) The *tensor outer product* between a Pth-order tensor $\mathbf{U} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_P}$ with entries $u_{i_1\cdots i_P}$ and a Qth-order tensor $\mathbf{V} \in \mathbb{R}^{J_1\times J_2\times\cdots\times J_Q}$ with entries $v_{j_1\cdots j_Q}$ is denoted by $\mathbf{W} = \mathbf{U}\circ\mathbf{V}$. The result $\mathbf{W} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_P\times J_1\times J_2\times\cdots\times J_Q}$ is a $(P+Q)$th-order tensor, whose entries are calculated by

$$w_{i_1\cdots i_P j_1\cdots j_Q} = u_{i_1\cdots i_P} \cdot v_{j_1\cdots j_Q}. \tag{6}$$

The major use of the tensor outer product is to construct a higher order tensor with several lower order tensors. For example, the tensor outer product between the vectors $\mathbf{a} \in \mathbb{R}^M$ and $\mathbf{b} \in \mathbb{R}^N$ is denoted by

$$\mathbf{T} = \mathbf{a} \circ \mathbf{b} \tag{7}$$

where the result $\mathbf{T}$ is a matrix in $\mathbb{R}^{M\times N}$ with entries $t_{ij} = a_i \cdot b_j$ for $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$. Now, we introduce one more vector $\mathbf{c} \in \mathbb{R}^Q$, where

$$\mathbf{S} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} = \mathbf{T} \circ \mathbf{c}. \tag{8}$$

Here, the result $\mathbf{S}$ is a third-order tensor with entries $s_{ijk} = a_i \cdot b_j \cdot c_k = t_{ij} \cdot c_k$ for $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, Q$.

2) The *n-mode product* between a tensor $\mathbf{U} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_P}$ and a matrix $\mathbf{V} \in \mathbb{R}^{\mathbf{J}\times\mathbf{I_n}}$ is denoted by $\mathbf{W} = \mathbf{U} \times_{\mathbf{n}} \mathbf{V} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_{n-1}\times J\times I_{n+1}\times\cdots\times I_P}$. Each element in $\mathbf{W}$ is defined as

$$w_{i_1 i_2\cdots i_{n-1} j i_{n+1}\cdots i_P} = \sum_{i_n=1}^{I_n} u_{i_1\cdots i_P} v_{j i_n} \tag{9}$$

where the main function is to adjust the dimension of a specific order. For example, in (9), the dimension of the nth-order of $\mathbf{U}$ is changed from I_n to J.

3) The *Kronecker product* of matrices $\mathbf{U} \in \mathbb{R}^{I\times J}$ and $\mathbf{V} \in \mathbb{R}^{P\times Q}$ is defined as

$$\mathbf{U} \otimes \mathbf{V} = \begin{bmatrix} u_{11}\mathbf{V} & u_{12}\mathbf{V} & \cdots & u_{1J}\mathbf{V} \\ u_{21}\mathbf{V} & u_{22}\mathbf{V} & \cdots & u_{2J}\mathbf{V} \\ \vdots & \vdots & \ddots & \vdots \\ u_{I1}\mathbf{V} & u_{I2}\mathbf{V} & \cdots & u_{IJ}\mathbf{V} \end{bmatrix} \tag{10a}$$

to generate an $IP \times JQ$ matrix.

4) The Khatri–Rao product between $\mathbf{U} \in \mathbb{R}^{I\times K}$ and $\mathbf{V} \in \mathbb{R}^{J\times K}$ is defined as

$$\mathbf{U} \odot \mathbf{V} = [\mathbf{u}_1 \otimes \mathbf{v}_1 \quad \mathbf{u}_2 \otimes \mathbf{v}_2 \quad \cdots \quad \mathbf{u}_K \otimes \mathbf{v}_K]. \tag{11}$$

5) The *Hadamard product* between $\mathbf{U} \in \mathbb{R}^{P\times Q}$ and $\mathbf{V} \in \mathbb{R}^{P\times Q}$ is defined as

$$\mathbf{U} * \mathbf{V} = \begin{bmatrix} u_{11}v_{11} & u_{12}v_{12} & \cdots & u_{1Q}v_{1Q} \\ u_{21}v_{21} & u_{22}v_{22} & \cdots & u_{2Q}v_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ u_{P1}v_{P1} & u_{P2}v_{P2} & \cdots & u_{PQ}v_{PQ} \end{bmatrix}. \tag{12}$$

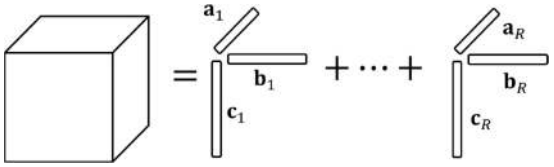

Fig. 5. CP decomposition of a third-order tensor.

3) Tensor Decomposition: Similar to the eigen-decomposition for matrix, tensor decomposition analyzes tensors via factorization. The CP decomposition is a widely used method, which factorizes a tensor into a sum of component rank-one tensors [24], [47]. For example, a third-order tensor $\mathbf{T} \in \mathbb{R}^{I\times J\times K}$ is decomposed into

$$\mathbf{T} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{13}$$

where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, $\mathbf{c}_r \in \mathbb{R}^K$, and R is a positive integer known as rank, which leads to the smallest number of rank-one tensors in the decomposition. The process of CP decomposition for a third-order tensor is illustrated in Fig. 5.

There are several extensions and alternatives of the CP decomposition. For example, the orthogonal-CP decomposition [26] decomposes the tensor using an orthogonal basis. For an Mth-order N-dimension tensor $\mathbf{T} \in \mathbb{R}^{\underbrace{N\times N\times\cdots\times N}_{M \text{ times}}}$, it can be decomposed by the orthogonal-CP decomposition as

$$\mathbf{T} = \sum_{r=1}^{R} \lambda_r \cdot \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(M)} \tag{14}$$

where $\lambda_r \geq 0$ and the orthogonal basis is $\mathbf{a}_r^{(i)} \in \mathbb{R}^N$ for $1 \leq i \leq M$. More specifically, the orthogonal-CP decomposition has a similar form to the eigen-decomposition when $M = 2$ and $\mathbf{T}$ is super-symmetric.

4) Tensor Spectrum: The eigenvalues and spectral space of tensors are significant topics in the tensor algebra. The research of the tensor spectrum has achieved great progress in recent years. It will take a large volume to cover all the properties of the tensor spectrum. Here, we just list some helpful and relevant literature. In particular, Lim [86] developed the theories of eigenvalues, eigenvectors, singular values, and singular vectors for tensors based on a constrained variational approach, such as the Rayleigh quotient. Qi [25], [85] presented a more complete discussion of tensor eigenvalues by defining two forms of tensor eigenvalues, i.e., the E-eigenvalue and the H-eigenvalue. Chang *et al.* [44] further extended the work of [25] and [85]. Other works, including [87] and [88], further developed the theory of tensor spectrum.

III. Definitions for Hypergraph Signal Processing

In this section, we introduce the core definitions used in our HGSP framework.

A. Algebraic Representation of Hypergraphs

The traditional GSP mainly relies on the representing matrix of a graph. Thus, an effective algebraic representation is also helpful in developing a novel HGSP framework. As we mentioned in Section II-C, tensor is an intuitive representation for high-dimensional graphs. In this section, we introduce the algebraic representation of hypergraphs based on tensors.

Similar to an adjacency matrix whose 2-D entries indicate whether and how two nodes are pairwise connected by a simple edge, we adopt an adjacency tensor whose entries indicate whether and how corresponding subsets of M nodes are connected by hyperedges to describe hypergraphs [20].

Definition 2 (Adjacency Tensor): A hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with N nodes and $m.c.e(\mathcal{H}) = M$ can be represented by an Mth-order N-dimension adjacency tensor $\mathbf{A} \in \mathbb{R}^{\underbrace{N\times N\times\cdots\times N}_{M \text{ times}}}$ defined as

$$\mathbf{A} = \left(a_{i_1 i_2 \cdots i_M}\right), \quad 1 \leq i_1, i_2, \ldots, i_M \leq N. \tag{15}$$

Suppose that $\mathbf{e}_l = \{\mathbf{v}_{l1}, \mathbf{v}_{l2}, \ldots, \mathbf{v}_{lc}\} \in \mathcal{E}$ is a hyperedge in $\mathcal{H}$ with the number of vertices $c \leq M$. Then, $\mathbf{e}_l$ is represented by all the elements $a_{p_1\cdots p_M}$'s in $\mathbf{A}$, where a subset of c indices from $\{p_1, p_2, \ldots, p_M\}$ are exactly the same as $\{l_1, l_2, \ldots, l_c\}$ and the other $M-c$ indices are picked from $\{l_1, l_2, \ldots, l_c\}$ randomly. More specifically, these elements $a_{p_1\cdots p_M}$'s describing $\mathbf{e}_l$ are calculated as

$$a_{p_1\cdots p_M} = c\left(\sum_{k_1,k_2,\ldots,k_c\geq 1, \sum_{i=1}^{c} k_i = M} \frac{M!}{k_1!k_2!\cdots k_c!}\right)^{-1}. \tag{16}$$

Meanwhile, the entries, which do not correspond to any hyperedge $\mathbf{e} \in \mathcal{E}$, are zeros.

Note that (16) enumerates all the possible combinations of c positive integers $\{k_1, \ldots, k_c\}$, whose summation satisfies $\sum_{i=1}^{c} k_i = M$. Obviously, when the hypergraph degrades to the normal graph with $c = M = 2$, the weights of edges are calculated as one, i.e., $a_{ij} = a_{ji} = 1$ for an edge $\mathbf{e} = (i, j) \in \mathcal{E}$. Then, the adjacency tensor is the same as the adjacency matrix. To understand the physical meaning of the adjacency tensor and its weight, we start with the M-uniform hypergraph with N nodes, where each hyperedge has exactly M nodes [53]. Since each hyperedge has an equal number of nodes, all hyperedges follow a consistent form to describe an M-lateral relationship with $m.c.e = M$. Obviously, such M-lateral relationships can be represented by an Mth-order tensor $\mathbf{A}$, where the entry $a_{i_1 i_2 \cdots i_M}$ indicates whether the nodes $\mathbf{v}_{i_1}, \mathbf{v}_{i_2}, \ldots, \mathbf{v}_{i_M}$ are in the same hyperedge, i.e., whether a hyperedge $\mathbf{e} = \{\mathbf{v}_{i_1}, \mathbf{v}_{i_2}, \ldots, \mathbf{v}_{i_M}\}$ exists. If the weight is nonzero, the hyperedge exists; otherwise, the hyperedge does not exist. Taking the 3-uniform hypergraph in Fig. 6(a) as an example, the hyperedge $\mathbf{e}_1$ is characterized by $a_{146} = a_{164} = a_{461} = a_{416} = a_{614} = a_{641} \neq 0$, the hyperedge $\mathbf{e}_2$ is characterized by $a_{237} = a_{327} = a_{732} = a_{723} = a_{273} = a_{372} \neq 0$, and $\mathbf{e}_3$ is represented by $a_{567} = a_{576} = a_{657} = a_{675} = a_{756} = a_{765} \neq 0$. All other entries in $\mathbf{A}$ are zero. Note that all the hyperedges in an M-uniform hypergraph has the same weight. Different hyperedges are distinguished by the indices of the entries. More specifically, similar to a_{ij} in the adjacency matrix implies the connection direction from node $\mathbf{v}_j$ to node $\mathbf{v}_i$ in GSP, an entry $a_{i_1 i_2 \cdots i_M}$ characterizes one direction of the hyperedge

Fig. 6. Examples of hypergraphs. (a) Three-uniform hypergraph $\mathcal{H}$ with seven nodes, three hyperedges, and $m.c.e(\mathcal{H}) = 3$, where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_7\}$ and $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. Three hyperedges are $\mathbf{e}_1 = \{\mathbf{v}_1, \mathbf{v}_4, \mathbf{v}_6\}$, $\mathbf{e}_2 = \{\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_7\}$, and $\mathbf{e}_3 = \{\mathbf{v}_5, \mathbf{v}_6, \mathbf{v}_7\}$. (b) General hypergraph $\mathcal{H}$ with seven nodes, three hyperedges, and $m.c.e(\mathcal{H}) = 3$, where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_7\}$ and $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. Three hyperedges are $\mathbf{e}_1 = \{\mathbf{v}_1, \mathbf{v}_4, \mathbf{v}_6\}$, $\mathbf{e}_2 = \{\mathbf{v}_2, \mathbf{v}_3\}$, and $\mathbf{e}_3 = \{\mathbf{v}_5, \mathbf{v}_6, \mathbf{v}_7\}$.

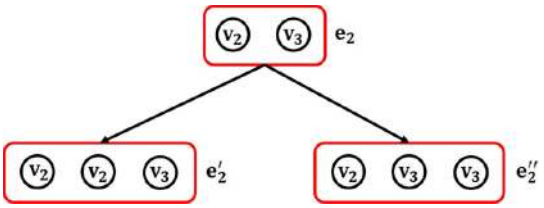

Fig. 7. Interpretation of generalizing $\mathbf{e}_2$ to hyperedges with $M = 3$.

$\mathbf{e} = \{\mathbf{v}_{i_1}, \mathbf{v}_{i_2}, \ldots, \mathbf{v}_{i_M}\}$ with node $\mathbf{v}_{i_M}$ as the source and node $\mathbf{v}_{i_1}$ as the destination.

However, for a general hypergraph, different hyperedges may contain different numbers of nodes. For example, in the hypergraph of Fig. 6(b), the hyperedge $\mathbf{e}_2$ only contains two nodes. How to represent the hyperedges with the number of nodes below $m.c.e = M$ may become an issue. To represent such hyperedge $\mathbf{e}_l = \{\mathbf{v}_{l_1}, \mathbf{v}_{l_2}, \ldots, \mathbf{v}_{l_c}\} \in \mathcal{E}$ with the number of vertices $c < M$ in an Mth-order tensor, we can use entries $a_{i_1 i_2 \cdots i_M}$, where a subset of c indices are the same as $\{l_1, \ldots, l_c\}$ (possibly a different order) and the other $M - c$ indices are picked from $\{l_1, \ldots, l_c\}$ randomly. This process can be interpreted as generalizing the hyperedge with c nodes to a hyperedge with M nodes by duplicating $M-c$ nodes from the set $\{\mathbf{v}_{l_1}, \ldots, \mathbf{v}_{l_c}\}$ randomly with possible repetitions. For example, the hyperedge $\mathbf{e}_2 = \{\mathbf{v}_2, \mathbf{v}_3\}$ in Fig. 6(b) can be represented by the entries $a_{233} = a_{323} = a_{332} = a_{322} = a_{223} = a_{232}$ in the third-order tensor $\mathbf{A}$, which could be interpreted as generalizing the original hyperedge with $c = 2$ to hyperedges with $M = 3$ nodes as Fig. 7. We can use (16) as a generalization coefficient of each hyperedge with respect to permutation and combination [20]. More specifically, for the adjacency tensor of the hypergraph in Fig. 6(b), the entries are calculated as $a_{146} = a_{164} = a_{461} = a_{416} = a_{614} = a_{641} = a_{567} = a_{576} = a_{657} = a_{675} = a_{756} = a_{765} = (1/2)$ and $a_{233} = a_{323} = a_{332} = a_{322} = a_{223} = a_{232} = (1/3)$, where the remaining entries are set to zeros. Note that the weight is smaller if the original hyperedge has fewer nodes in Fig. 6(b). More generally, based on the definition of adjacency tensor and (16), we can easily obtain the following property regarding the hyperedge weight.

Property 1: Given two hyperedges $\mathbf{e}_i = \{\mathbf{v}_1, \ldots, \mathbf{v}_I\}$ and $\mathbf{e}_j = \{\mathbf{v}_1, \ldots, \mathbf{v}_J\}$, the edgeweight $w(\mathbf{e}_i)$ of $\mathbf{e}_i$ is different from the edgeweight $w(\mathbf{e}_j)$ of $\mathbf{e}_j$ in the adjacency tensor $\mathbf{A}$, i.e., $w(\mathbf{e}_i) \neq w(\mathbf{e}_j)$, if $I \neq J$. Moreover, $w(\mathbf{e}_i) = w(\mathbf{e}_j)$ iff $I = J$.

This property can help identify the length of each hyperedge based on the weights in the adjacency tensor. Moreover, the edgeweights of two hyperedges with the same number of nodes are the same. Different hyperedges with the same number of nodes are distinguished by their indices of entries in an adjacency tensor.

The degree $d(\mathbf{v}_i)$, of a vertex $\mathbf{v}_i \in \mathcal{V}$, is the number of hyperedges containing $\mathbf{v}_i$, i.e.,

$$d(\mathbf{v}_i) = \sum_{j_1, j_2, \ldots, j_{M-1}=1}^{N} a_{ij_1 j_2 \cdots j_{M-1}}. \tag{17}$$

Then, the Laplacian tensor of the hypergraph $\mathcal{H}$ is defined as follows [20].

Definition 3 (Laplacian Tensor): Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with N nodes and $m.c.e(\mathcal{H}) = M$, the Laplacian tensor is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \in \mathbb{R}^{\underbrace{N \times N \times \cdots \times N}_{M \text{ times}}} \tag{18}$$

which is an Mth-order N-dimension tensor. Here, $\mathbf{D} = (d_{i_1 i_2 \cdots i_M})$ is also an Mth-order N-dimension super-diagonal tensor with nonzero elements of $d_{\underbrace{ii \ldots i}_{M \text{ times}}} = d(\mathbf{v}_i)$.

We see that both the adjacency and Laplacian tensors of a hypergraph $\mathcal{H}$ are super-symmetric. Moreover, when $m.c.e(\mathcal{H}) = 2$, they have similar forms to the adjacency and Laplacian matrices of undirected graphs, respectively. Similar to GSP, we use an Mth-order N-dimension tensor $\mathbf{F}$ as a general representation of a given hypergraph $\mathcal{H}$ for convenience. As the adjacency tensor is more general, the representing tensor $\mathbf{F}$ refers to the adjacency tensor in this article unless specified otherwise.

B. Hypergraph Signal and Signal Shifting

Based on the tensor representation of hypergraphs, we now provide definitions for the hypergraph signal. In the traditional GSP, each signal element is related to one node in the graph. Thus, the graph signal in GSP is defined as an N-length vector if there are N nodes in the graph. Recall that the representing matrix of a normal graph can be treated as a graph filter, for which the basic form of the filtered signal is defined in (1).

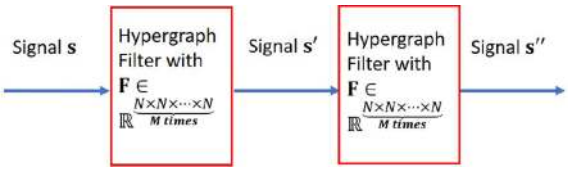

Fig. 8. Signals in a polynomial filter.

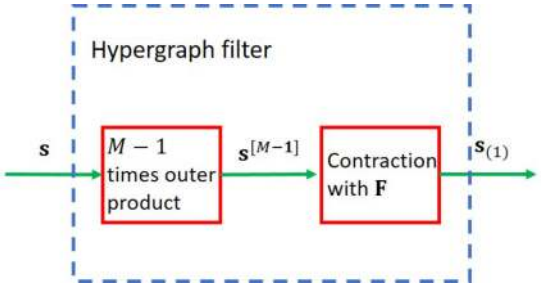

Fig. 9. Diagram of hypergraph shifting.

Thus, we could extend the definitions of the graph signal and signal shifting from the traditional GSP to HGSP based on the tensor-based filter implementation.

In HGSP, we also relate signal element to one node in the hypergraph. Naturally, we can define the original signal as an N-length vector if there are N nodes. Similarly, as in GSP, we define the hypergraph shifting based on the representing tensor $\mathbf{F}$. However, since tensor $\mathbf{F}$ is of Mth-order, we need an $(M-1)$th-order signal tensor to work with the hypergraph filter $\mathbf{F}$, such that the filtered signal is also an N-length vector as the original signal. For example, for a two-step polynomial filter shown as Fig. 8, the signals $\mathbf{s}, \mathbf{s}', \mathbf{s}''$ should all be in the same dimension and order. For the input and output signals in an HGSP system to have a consistent form, we define an alternative form of the hypergraph signal as below.

Definition 4 (Hypergraph Signal): For a hypergraph $\mathcal{H}$ with N nodes and $m.c.e(\mathcal{H}) = M$, an alternative form of hypergraph signal is an $(M-1)$th-order N-dimension tensor $\mathbf{s}^{[M-1]}$ obtained from $(M-1)$ times outer product of the original signal $\mathbf{s} = [s_1 \quad s_2 \quad \cdots \quad s_N]^T$, i.e.,

$$\mathbf{s}^{[M-1]} = \underbrace{\mathbf{s} \circ \cdots \circ \mathbf{s}}_{M-1 \text{ times}} \tag{19}$$

where each entry in position $(i_1, i_2, \ldots, i_{M-1})$ equals the product $s_{i_1} s_{i_2} \cdots s_{i_{M-1}}$.

Note that the above hypergraph signal comes from the original signal. They are different forms of the same signal, which reflect the signal properties in different dimensions. For example, a second-order hypergraph signal highlights the properties of the 2-D signal components $s_i s_j$ while the original signal directly emphasizes more about the one-dimension properties. We will discuss in detail on the relationship between the hypergraph signal and the original signal in Section III-D.

With the definition of hypergraph signals, let us define the original domain of signals for convenience before we step into the signal shifting. Similarly as that the signals lie in the time domain for DSP, we have the following definition of the hypergraph vertex domain.

Definition 5 (Hypergraph Vertex Domain): A signal lies in the hypergraph vertex domain if it resides on the structure of a hypergraph in the HGSP framework.

The hypergraph vertex domain is a counterpart of time domain in HGSP. The signals are analyzed based on the structure among vertices in a hypergraph.

Next, we discuss how the signals shift on the given hypergraph. Recall that in GSP, the signal shifting is defined by the product of the representing matrix $\mathbf{F_M} \in \mathbb{R}^{N \times N}$ and the signal vector $\mathbf{s} \in \mathbb{R}^N$, i.e., $\mathbf{s}' = \mathbf{F_M s}$. Similarly, we define the hypergraph signal shifting based on its tensor $\mathbf{F}$ and the hypergraph signal $\mathbf{s}^{[M-1]}$.

Definition 6 (Hypergraph Shifting): The basic shifting filter of hypergraph signals is defined as the direct contraction between the representing tensor $\mathbf{F}$ and the hypergraph signals $\mathbf{s}^{[M-1]}$, i.e.,

$$\mathbf{s}_{(1)} = \mathbf{F}\mathbf{s}^{[M-1]} \tag{20}$$

where each element of the filter output is given by

$$\left(\mathbf{s}_{(1)}\right)_i = \sum_{j_1,\ldots,j_{M-1}=1}^{N} f_{ij_1\cdots j_{M-1}} s_{j_1} s_{j_2} \cdots s_{j_{M-1}}. \tag{21}$$

Since the hypergraph signal contracts with the representing tensor in $M-1$ order, the one-time filtered signal $\mathbf{s}_{(1)}$ is an N-length vector, which has the same dimension as the original signal. Thus, the block diagram of a hypergraph filter with $\mathbf{F}$ can be shown as Fig. 9.

Let us now consider the functionality of the hypergraph filter, as well as the physical insight of the hypergraph shifting. In GSP, the functionality of the filter $\mathbf{F_M}$ is simply to shift the signals along the link directions. However, interactions inside the hyperedge are more complex as it involves more than two nodes. In (21), we see that the filtered signal in $\mathbf{v}_i$ equals the summation of the shifted signal components in all hyperedges containing node $\mathbf{v}_i$, where $f_{ij_1\cdots j_{M-1}}$ is the weight for each involved hyperedge and $\{s_{j_1}, \ldots, s_{j_{M-1}}\}$ are the signals in the generalized hyperedges excluding s_i. Clearly, the hypergraph shifting multiplies signals in the same hyperedge of node $\mathbf{v}_i$ together before delivering the shift to a certain node $\mathbf{v}_i$. Taking the hypergraph in Fig. 6(a) as an example, node $\mathbf{v}_7$ is included in two hyperedges, $\mathbf{e}_2 = \{\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_7\}$ and $\mathbf{e}_3 = \{\mathbf{v}_5, \mathbf{v}_6, \mathbf{v}_7\}$. According to (21), the shifted signal in node $\mathbf{v}_7$ is calculated as

$$s_7 = f_{732} \times s_2 s_3 + f_{723} \times s_2 s_3 + f_{756} \times s_5 s_6 + f_{765} \times s_5 s_6 \tag{22}$$

where $f_{732} = f_{723}$ is the weight of the hyperedge $\mathbf{e}_2$ and $f_{756} = f_{765}$ is the weight of the hyperedge $\mathbf{e}_3$ in the adjacency tensor $\mathbf{F}$.

As the entry a_{ji} in the adjacency matrix of a normal graph indicates the link direction from the node $\mathbf{v}_i$ to the node $\mathbf{v}_j$, the entry $f_{i_1\cdots i_M}$ in the adjacency tensor similarly indicates the order of nodes in a hyperedge as $\{\mathbf{v}_{i_M}, \mathbf{v}_{i_{M-1}}, \ldots, \mathbf{v}_{i_1}\}$, where $\mathbf{v}_{i_1}$ is the destination and $\mathbf{v}_{i_M}$ is the source. Thus, the shifting

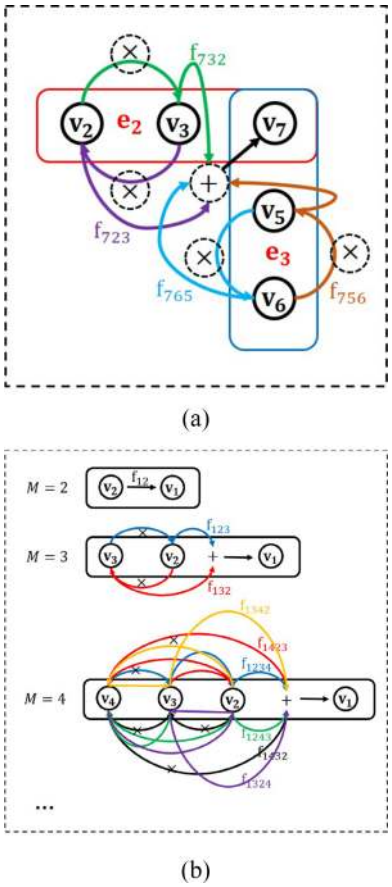

Fig. 10. Diagram of signal shifting. (a) Example of signal shifting to node $\mathbf{v}_7$. Different colors of arrows show the different directions of shifting; "×" refers to multiplication and "+" refers to summation. (b) Diagram of signal shifting to node $\mathbf{v}_1$ in an M-way hyperedge. Different colors refer to different shifting directions.

by (22) could be interpreted as shown in Fig. 10(a). Since there are two possible directions from nodes $\{\mathbf{v}_2, \mathbf{v}_3\}$ to node $\mathbf{v}_7$ in $\mathbf{e}_2$, there are two components shifted to $\mathbf{v}_7$, i.e., the first two terms in (22). Similarly, there are also two components shifted by the hyperedge $\mathbf{e}_3$, i.e., the last two terms in (22). To illustrate the hypergraph shifting more explicitly, Fig. 10(b) shows a diagram of signal shifting to a certain node in an M-way hyperedge. From Fig. 10(b), we see that the graph shifting in GSP is a special case of the hypergraph shifting, where $M = 2$. Moreover, there are $K = (M-1)!$ possible directions for the shifting to one specific node in an M-way hyperedge.

C. Hypergraph Spectrum Space

We now provide the definitions of the hypergraph Fourier space, i.e., the hypergraph spectrum space. In GSP, the graph Fourier space is defined as the eigenspace of its representing matrix [5]. Similarly, we define the Fourier space of HGSP based on the representing tensor $\mathbf{F}$ of a hypergraph, which characterizes the hypergraph structure and signal shifting. For an Mth-order N-dimension tensor $\mathbf{F}$, we can apply the orthogonal-CP decomposition [26] to write

$$\mathbf{F} = \sum_{r=1}^{R} \lambda_r \cdot \mathbf{f}_r^{(1)} \circ \cdots \circ \mathbf{f}_r^{(M)} \tag{23}$$

with basis $\mathbf{f}_r^{(i)} \in \mathbb{R}^N$ for $1 \le i \le M$ and $\lambda_r \ge 0$. Since $\mathbf{F}$ is super-symmetric [25], i.e., $\mathbf{f}_r = \mathbf{f}_r^{(1)} = \mathbf{f}_r^{(2)} = \cdots = \mathbf{f}_r^{(M)}$, we have

$$\mathbf{F} = \sum_{r=1}^{R} \lambda_r \cdot \underbrace{\mathbf{f}_r \circ \cdots \circ \mathbf{f}_r}_{M \text{ times}}. \tag{24}$$

Generally, we have the rank $R \le N$ in a hypergraph. We will discuss how to construct the remaining $\mathbf{f}_i$, $R < i \le N$, for the case of $R < N$ later in Section III-F.

Now, by plugging (24) into (20), the hypergraph shifting can be written with the N basis $\mathbf{f}_i$'s as

$$\mathbf{s}_{(1)} = \mathbf{F}\mathbf{s}^{[M-1]} \tag{25a}$$

$$= \left(\sum_{r=1}^{N} \lambda_r \cdot \underbrace{\mathbf{f}_r \circ \cdots \circ \mathbf{f}_r}_{M \text{ times}} \right) \left(\underbrace{\mathbf{s} \circ \cdots \circ \mathbf{s}}_{M-1 \text{ times}} \right) \tag{25b}$$

$$= \sum_{r=1}^{N} \lambda_r \mathbf{f}_r \underbrace{<\mathbf{f}_r, \mathbf{s}> \cdots <\mathbf{f}_r, \mathbf{s}>}_{M-1 \text{ times}} \tag{25c}$$

$$= \underbrace{\begin{bmatrix} \mathbf{f}_1 & \cdots & \mathbf{f}_N \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix}}_{\text{iHGFT and filter in Fourier space}} \times \underbrace{\begin{bmatrix} (\mathbf{f}_1^T \mathbf{s})^{M-1} \\ \vdots \\ (\mathbf{f}_N^T \mathbf{s})^{M-1} \end{bmatrix}}_{\text{HGFT of the hypergraph signal}} \tag{25d}$$

where $<\mathbf{f}_r, \mathbf{s}> = (\mathbf{f}_i^T \mathbf{s})$ is the inner product between $\mathbf{f}_r$ and $\mathbf{s}$, and $(\cdot)^{M-1}$ is $(M-1)$th power.

From (25d), we see that the shifted signal in HGSP is in a similar decomposed to (3) and (4) for GSP. The first two parts in (25d) work like $\mathbf{V}_\mathbf{M}^{-1}\Lambda$ of the GSP eigen-decomposition, which could be interpreted as inverse Fourier transform and filter in the Fourier space. The third part can be understood as the HGFT of the original signal. Hence, similarly as in GSP, we can define the hypergraph Fourier space and Fourier transform based on the orthogonal-CP decomposition of $\mathbf{F}$.

Definition 7 (Hypergraph Fourier Space and Fourier Transform): The hypergraph Fourier space of a given hypergraph $\mathcal{H}$ is defined as the space consisting of all orthogonal-CP decomposition basis $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$. The frequencies are defined with respect to the eigenvalue coefficients λ_i, $1 \le i \le N$. The HGFT of hypergraph signals is defined as

$$\hat{\mathbf{s}} = \mathcal{F}_C\left(\mathbf{s}^{[M-1]}\right) \tag{26a}$$

$$= \begin{bmatrix} (\mathbf{f}_1^T \mathbf{s})^{M-1} \\ \vdots \\ (\mathbf{f}_N^T \mathbf{s})^{M-1} \end{bmatrix}. \tag{26b}$$

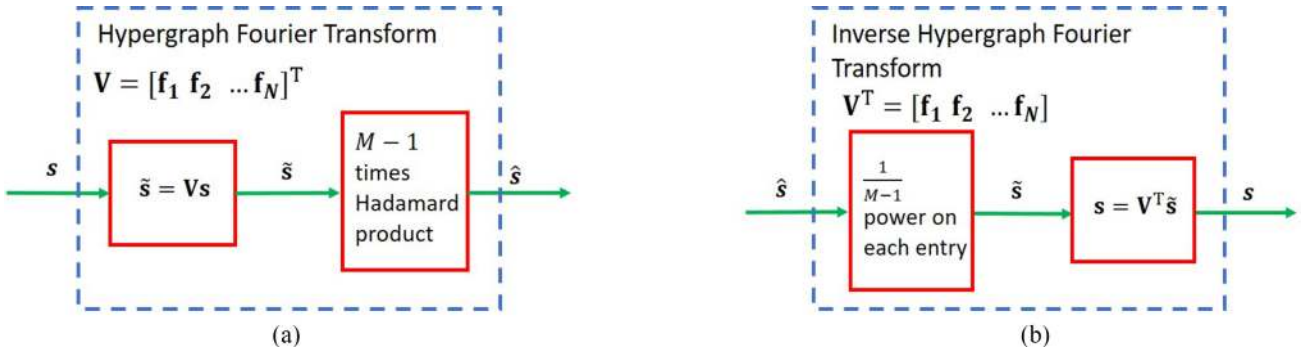

Fig. 11. Diagram of (a) HGFT and (b) iHGFT.

Compared to GSP, if $M = 2$, the HGFT has the same form as the traditional GFT. In addition, since $\mathbf{f}_r$ is the orthogonal basis, we have

$$\mathbf{F}\mathbf{f}_r^{[M-1]} = \sum \lambda_i \mathbf{f}_i \left(\mathbf{f}_i^T \mathbf{f}_r\right)^{M-1} = \lambda_r \mathbf{f}_r. \tag{27}$$

According to [25], a vector $\mathbf{x}$ is an E-eigenvector of an Mth-order tensor $\mathbf{A}$, if $\mathbf{A}\mathbf{x}^{[M-1]} = \lambda\mathbf{x}$ exists for a constant λ. Then, we obtain the following property of the hypergraph spectrum.

Property 2: The hypergraph spectrum pair $(\lambda_r, \mathbf{f}_r)$ is an E-eigenpair of the representing tensor $\mathbf{F}$.

Recall that the spectrum space of GSP is the eigenspace of the representing matrix $\mathbf{F_M}$. Property 2 shows that HGSP has a consistent definition in the spectrum space as that for GSP.

D. Relationship Between Hypergraph Signal and Original Signal

With HGFT defined, let us discuss more about the relationship between the hypergraph signal and the original signal in the Fourier space to understand the HGFT better. From (26b), the hypergraph signal in the Fourier space is written as

$$\hat{\mathbf{s}} = \begin{bmatrix} \left(\mathbf{f}_1^T \mathbf{s}\right)^{M-1} \\ \vdots \\ \left(\mathbf{f}_N^T \mathbf{s}\right)^{M-1} \end{bmatrix} \tag{28}$$

which can be further decomposed as

$$\hat{\mathbf{s}} = \underbrace{\left(\begin{bmatrix} \mathbf{f}_1^T \\ \vdots \\ \mathbf{f}_N^T \end{bmatrix} \mathbf{s}\right) * \left(\begin{bmatrix} \mathbf{f}_1^T \\ \vdots \\ \mathbf{f}_N^T \end{bmatrix} \mathbf{s}\right) * \cdots * \left(\begin{bmatrix} \mathbf{f}_1^T \\ \vdots \\ \mathbf{f}_N^T \end{bmatrix} \mathbf{s}\right)}_{M-1 \text{ times}} \tag{29}$$

where $*$ denotes Hadamard product.

From (29), we see that the hypergraph signal in the hypergraph Fourier space is $M-1$ times Hadamard product of a component consisting of the hypergraph Fourier basis and the original signal. More specifically, this component works as the original signal in the hypergraph Fourier space, which is defined as

$$\tilde{\mathbf{s}} = \mathbf{V}\mathbf{s} \tag{30}$$

where $\mathbf{V} = [\mathbf{f}_1 \quad \mathbf{f}_2 \quad \cdots \quad \mathbf{f}_N]^T$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

Recall the definitions of the hypergraph signal and vertex domain in Section III-B, we have the following property.

Property 3: The hypergraph signal is the $M-1$ times tensor outer product of the original signal in the hypergraph vertex domain, and the $M-1$ times Hadamard product of the original signal in the hypergraph frequency domain.

Then, we could establish a connection between the original signal and the hypergraph signal in the hypergraph Fourier domain by the HGFT and inverse HGFT (iHGFT) as shown in Fig. 11. Such a relationship leads to some interesting properties and makes the HGFT implementation more straightforward, which will be further discussed in Sections III-F and III-G, respectively.

E. Hypergraph Frequency

As we now have a better understanding of the hypergraph Fourier space and Fourier transform, we can discuss more about the hypergraph frequency and its order. In GSP, the graph frequency is defined with respect to the eigenvalues of the representing matrix $\mathbf{F_M}$ and ordered by the total variation [5]. Similarly, in HGSP, we define the frequency relative to the coefficients λ_i from the orthogonal-CP decomposition. We order them by the total variation of frequency components $\mathbf{f}_i$ over the hypergraph. The total variation of a general signal component over a hypergraph is defined as follows.

Definition 8 (Total Variation Over Hypergraph): Given a hypergraph $\mathcal{H}$ with N nodes and the normalized representing tensor $\mathbf{F}^{\text{norm}} = (1/\lambda_{\max})\mathbf{F}$, together with the original signal $\mathbf{s}$, the total variation over the hypergraph is defined as the total differences between the nodes and their corresponding neighbors in the perspective of shifting, i.e.,

$$\mathbf{TV}(\mathbf{s}) = \sum_{i=1}^{N} \left| s_i - \sum_{j_1,\ldots,j_{M-1}=1}^{N} F^{\text{norm}}_{ij_1\cdots j_{M-1}} s_{j_1} \cdots s_{j_{M-1}} \right| \tag{31a}$$

$$= \left\| \mathbf{s} - \mathbf{F}^{\text{norm}} \mathbf{s}^{[M-1]} \right\|_1. \tag{31b}$$

We adopt the l_1-norm here only as an example of defining the total variation. Other norms may be more suitable depending on specific applications. Now, with the definition of total variation over hypergraphs, the frequency in HGSP is ordered by the total variation of the corresponding frequency component $\mathbf{f}_r$, i.e.,

$$\mathbf{TV}(f_r) = \left\| \mathbf{f}_r - \mathbf{f}^{\text{norm}}_{r(1)} \right\|_1 \tag{32}$$

where $\mathbf{f}_{r(1)}^{\text{norm}}$ is the output of one-time shifting for $\mathbf{f}_r$ over the normalized representing tensor.

From (31a), we see that the total variation describes how much the signal component changes from a node to its neighbors over the hypergraph shifting. Thus, we have the following definition of hypergraph frequency.

Definition 9 (Hypergraph Frequency): Hypergraph frequency describes how oscillatory the signal component is with respect to the given hypergraph. A frequency component $\mathbf{f}_r$ is associated with a higher frequency if the total variation of this frequency component is larger.

Note that the physical meaning of graph frequency was stated in GSP [2]. Generally, the graph frequency is highly related to the total variation of the corresponding frequency component. Similarly, the hypergraph frequency also relates to the corresponding total variation. We will discuss more about the interpretation of the hypergraph frequency and its relationships with DSP and GSP later in Section IV-A, to further consolidate our hypergraph frequency definition.

Based on the definition of total variation, we describe one important property of $\mathbf{TV}(\mathbf{f}_r)$ in the following theorem.

Theorem 1: Define a supporting matrix

$$\mathbf{P_s} = \frac{1}{\lambda_{\max}}\begin{bmatrix}\mathbf{f}_1 & \cdots & \mathbf{f}_N\end{bmatrix}\begin{bmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_N\end{bmatrix}\begin{bmatrix}\mathbf{f}_1^T \\ \vdots \\ \mathbf{f}_N^T\end{bmatrix}. \quad (33)$$

With the normalized representing tensor $\mathbf{F}^{\text{norm}} = (1/\lambda_{\max})\mathbf{F}$, the total variation of hypergraph spectrum $\mathbf{f}_r$ is calculated as

$$\mathbf{TV}(\mathbf{f_r}) = \left\|\mathbf{f}_r - \mathbf{f}_{r(1)}^{\text{norm}}\right\|_1 \quad (34a)$$

$$= \|\mathbf{f}_r - \mathbf{P_s}\mathbf{f}_r\|_1 \quad (34b)$$

$$= \left|1 - \frac{\lambda_r}{\lambda_{\max}}\right|. \quad (34c)$$

Moreover, $\mathbf{TV}(\mathbf{f}_i) > \mathbf{TV}(\mathbf{f}_j)$ iff $\lambda_i < \lambda_j$.

Proof: For hypergraph signals, the output of one-time shifting of $\mathbf{f}_r$ is calculated as

$$\mathbf{f}_{r(1)} = \sum_{i=1}^{N} \lambda_i \mathbf{f}_i \left(\mathbf{f}_i^T \mathbf{f}_r\right)^{M-1} = \lambda_r \mathbf{f}_r. \quad (35)$$

Based on the normalized $\mathbf{F}^{\text{norm}}$, we have $\mathbf{f}_{r(1)}^{\text{norm}} = (\lambda_r/\lambda_{\max})\mathbf{f}_r$. It is therefore easy to obtain (34c) from (34a). To obtain (34b), we have

$$\mathbf{P_s}\mathbf{f}_r = \sum_{i=1}^{N} \lambda_i \mathbf{f}_i \left(\mathbf{f}_i^T \mathbf{f}_r\right) = \frac{\lambda_r}{\lambda_{\max}}\mathbf{f}_r. \quad (36)$$

It is clear that (34b) is the same as (34c).

Since λ is a real and non-negative, we have

$$\mathbf{TV}(f_i) - \mathbf{TV}\left(f_j\right) = \frac{\lambda_j - \lambda_i}{\lambda_{\max}}. \quad (37)$$

Obviously, $\mathbf{TV}(\mathbf{f}_i) > \mathbf{TV}(\mathbf{f}_j)$ iff $\lambda_i < \lambda_j$. ■

Theorem 1 shows that the supporting matrix $\mathbf{P_s}$ can help us apply the total variation more efficiently in some real applications. Moreover, it provides the order of frequency according to the coefficients λ_i's with the following property.

Property 4: A smaller λ is related to a higher frequency in the hypergraph Fourier space, where its corresponding spectrum basis is called a high frequency component.

F. Signals With Limited Spectrum Support

With the order of frequency, we define the bandlimited signals as follows.

Definition 10 (Bandlimited Signal): Order the coefficients as $\lambda = [\lambda_1 \quad \cdots \quad \lambda_N]$, where $\lambda_1 \geq \cdots \geq \lambda_N \geq 0$, together with their corresponding $\mathbf{f}_r$'s. A hypergraph signal $\mathbf{s}^{[M-1]}$ is defined as K-bandlimited if the HGFT transformed signal $\hat{\mathbf{s}} = [\hat{s}_1, \ldots, \hat{s}_N]^{\mathrm{T}}$ has $\hat{s}_i = 0$ for all $i \geq K$, where $K \in \{1, 2, \ldots, N\}$. The smallest K is defined as the bandwidth and the corresponding boundary is defined as $W = \lambda_K$.

Note that a larger λ_i corresponds to a lower frequency as we mentioned in Property 4. Then, the frequency are ordered from low to high in the definition above. Moreover, we use the index K instead of the coefficient value λ to define the bandwidth for the following reasons.

1) Identical λ's in two different hypergraphs do not refer to the same frequency. Since each hypergraph has its own adjacency tensor and spectrum space, the comparison of multiple spectrum pairs $(\lambda_i, \mathbf{f}_i)$'s is only meaningful within the same hypergraph. Moreover, there exists a normalization issue in the decomposition of different adjacency tensors. Thus, it is not meaningful to compare λ_k's across two different hypergraphs.
2) Since λ_k values are not continuous over k, different frequency cutoffs of λ may lead to the same bandlimited space. For example, suppose that $\lambda_k = 0.5$ and $\lambda_{k+1} = 0.8$. Then, $\lambda = 0.6$ and $\lambda' = 0.7$ would lead to the same cutoff in the frequency space, which makes bandwidth definition nonunique.

As we discussed in Section III-D, the hypergraph signal is the Hadamard product of the original signal in the frequency domain. Then, we have the following property of bandwidth.

Property 5: The bandwidth K is the same, based on the HGFT of the hypergraph signals $\hat{\mathbf{s}}$ and that of the original signals $\tilde{\mathbf{s}}$.

This property allows us to analyze the spectrum support of the hypergraph signal by looking into the original signal with lower complexity. Recall that we can add $\mathbf{f}_i$ by using zero coefficients λ_i when $R < N$ as mentioned in Section III-C. The added basis should not affect the HGFT signals in Fourier space. According to the structure of bandlimited signal, we need the added $\mathbf{f}_i$ could meet the following conditions: 1) $\mathbf{f}_i \perp \mathbf{f}_p$ for $p \neq i$; 2) $\mathbf{f}_i^T \cdot \mathbf{s} \to 0$; and 3) $|\mathbf{f}_i| = 1$.

G. Implementation and Complexity

We now consider the implementation and complexity issues of HGFT. Similar to GFT, the process of HGFT consists of two steps: 1) decomposition and 2) execution. The decomposition is to calculate the hypergraph spectrum basis, and the execution transforms signals from the hypergraph vertex domain into the spectrum domain.

1) The calculation of the spectrum basis by the orthogonal-CP decomposition is an important preparation step for

HGFT. A straightforward algorithm would decompose the representing tensor $\mathbf{F}$ with the spectrum basis $\mathbf{f}_i$'s and coefficients λ_i's as in (24). Efficient tensor decomposition is an active topic in both fields of mathematics and engineering. There are a number of methods for CP decomposition in the literature. In [54] and [58], motivated by the spectral theorem for real symmetric matrices, orthogonal-CP decomposition algorithms for symmetric tensors are developed based on the polynomial equations. Afshar *et al.* [26] proposed a more general decomposition algorithm for spatio-temporal data. Other works, including [55]–[57], tried to develop faster decomposition methods for signal processing and big data applications. The rapid development of tensor decomposition and the advancement of computation ability will benefit the efficient derivation of hypergraph spectrum.

2) The execution of HGFT with a known spectrum basis is defined in (26b). According to (29), the HGFT of hypergraph signal is an $M-1$ times Hadamard product of the original signal in the hypergraph spectrum space. This relationship can help execute HGFT and iHGFT of hypergraph signals more efficiently by applying matrix operations on the original signals. Clearly, the complexity of calculating the original signals in the frequency domain $\tilde{\mathbf{s}} = \mathbf{V}\mathbf{s}$ is $O(N^2)$. In addition, since the computation complexity of the power function $x^{(M-1)}$ could be $O(\log(M-1))$ and each vector has N entries, the complexity of calculating the $M-1$ times Hadamard product is $O(N\log(M-1))$. Thus, the complexity of general HGFT implementation is $O(N^2+N\log(M-1))$.

IV. Discussions and Interpretations

In this section, we focus on the insights and physical meaning of frequency to help interpret the hypergraph spectrum space. We also consider the relationships between HGSP and other existing works to better understand the HGSP framework.

A. Interpretation of Hypergraph Spectrum Space

We are interested in an intuitive interpretation of the hypergraph frequency and its relations with the DSP and GSP frequencies. We start with the frequency and the total variation in DSP. In DSP, the discrete Fourier transform (DFT) of a sequence s_n is given by $\hat{s}_k = \sum_{n=0}^{N-1} s_n e^{-j(2\pi kn/N)}$ and the frequency is defined as $\nu_n = (n/N)$, $n = 0, 1, \ldots, N-1$. From [38], we can easily summarize the following conclusions.

1) $\nu_n : 1 < n < (N/2) - 1$ corresponds to a continuous time signal frequency $(n/N)f_s$.
2) $\nu_n : (N/2)+1 < n < N-1$ corresponds to a continuous time signal frequency $-(1-[n/N])f_s$.
3) $\nu_{(N/2)}$ corresponds to $f_s/2$.
4) $n = 0$ corresponds to frequency 0.

Here, f_s is the critical sampling frequency. In traditional DFT, we generate the Fourier transform $\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x)e^{-2\pi jx\omega}dx$ at each discrete frequency $(n/N)f_s$, $n = -(N/2)+1, -(N/2)+2, \ldots, (N/2)-1, (N/2)$. The highest and lowest frequencies correspond to $n = N/2$ and $n = 0$, respectively. Note that n varies from $-(N/2)+1$ to $(N/2)$ here. Since $e^{-j2\pi k(n/N)} = e^{-j2\pi k(n+N/N)}$, we can let n vary from 0 to $N-1$ and cover the complete period. Now, n varies in exact correspondence to ν_n, and the aforementioned conclusions are drawn. The highest frequency occurs at $n = (N/2)$.

The total variation in DSP is defined as the differences among the signals over time [59], i.e.,

$$\mathbf{TV}(\mathbf{s}) = \sum_{n=0}^{N-1} \left| s_n - s_{(n-1 \mod N)} \right| \tag{38a}$$

$$= \|\mathbf{s} - \mathbf{C_N}\mathbf{s}\|_{\mathbf{1}} \tag{38b}$$

where

$$\mathbf{C_N} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \tag{39}$$

When we perform the eigen-decomposition of $\mathbf{C_N}$, we see that the eigenvalues are $\lambda_n = e^{-j(2\pi n/N)}$ with eigenvector $\mathbf{f}_n$, $0 \le n \le N-1$. More specifically, the total variation of the frequency component $\mathbf{f}_n$ is calculated as

$$\mathbf{TV}(\mathbf{f}_n) = \left| 1 - e^{j\frac{2\pi n}{N}} \right| \tag{40}$$

which increases with n for $n \le (N/2)$ before decreasing with n for $(N/2) < n \le N-1$.

Obviously, the total variations of frequency components have a one-to-one correspondence to frequencies in the order of their values. If the total variation of a frequency component is larger, the corresponding frequency with the same index n is higher. It also has a clear physical meaning, i.e., a higher frequency component changes faster over time, which implies a larger total variation. Thus, we could also use the total variation of a frequency component to characterize its frequency in DSP.

Let us now consider the total variation and frequency in GSP, where the signals are analyzed in the graph vertex domain instead of the time domain. Similar to the fact that the frequency in DSP describes the rate of signal changes over time, the frequency in GSP illustrates the rate of signal changes over vertex [5]. Likewise, the total variation of the graph Fourier basis defined according to the adjacency matrix $\mathbf{F_M}$ could be used to characterize each frequency. Since GSP handles signals in the graph vertex domain, the total variation of GSP is defined as the differences between all the nodes and their neighbors, i.e.,

$$\mathbf{TV}(\mathbf{s}) = \sum_{n=1}^{N} \left| s_n - \sum_m F_{M_{nm}}^{\text{norm}} s_m \right| \tag{41a}$$

$$= \left\| \mathbf{s} - \mathbf{F_M}^{\text{norm}}\mathbf{s} \right\|_1 \tag{41b}$$

where $\mathbf{F_M}^{\text{norm}} = (1/|\lambda_{\max}|)\mathbf{F_M}$. If the total variation of the frequency component $\mathbf{f_{M}}_i$ is larger, it means the change over

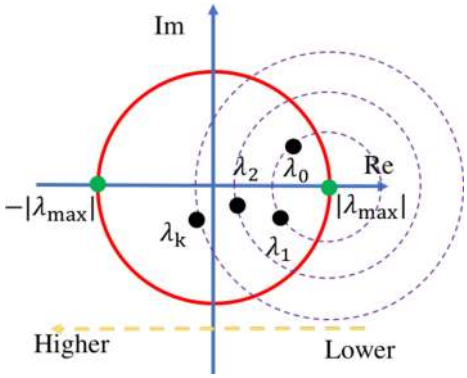

Fig. 12. Example of frequency order in GSP for complex eigenvalues.

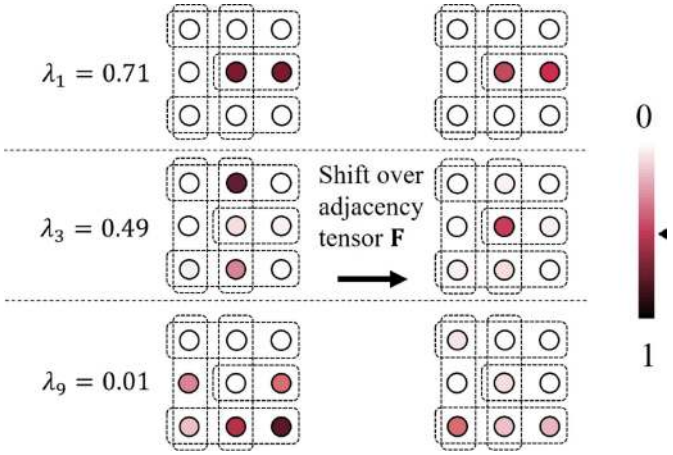

Fig. 13. Frequency components in a hypergraph with nine nodes and five hyperedges. The left panel shows the frequency components before shifting, and the right panel shows the frequency components after shifting. The values of signals are illustrated by colors. Higher frequency components imply larger changes across two panels.

the graph between neighborhood vertices is faster, which indicates a higher graph frequency. Note that once the graph is undirected, i.e., the eigenvalues are real numbers, the frequency decreases with the increase of the eigenvalue similar as HGSP in Section III-E; otherwise, if the graph is undirected, i.e., the eigenvalues are complex, the frequency changes as shown in Fig. 12, which has consistency with the changing pattern of DSP frequency [5].

We now turn to our HGSP framework. Like GSP, HGSP analyzes signals in the hypergraph vertex domain. Different from normal graphs, each hyperedge in HGSP connects more than two nodes. The neighbors of a vertex $\mathbf{v}_i$ include all the nodes in the hyperedges containing $\mathbf{v}_i$. For example, if there exists a hyperedge $\mathbf{e}_1 = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, nodes $\mathbf{v}_2$ and $\mathbf{v}_3$ are both neighbors of node $\mathbf{v}_1$. As we mentioned in Section III-E, the total variation of HGSP is defined as the difference between continuous signals over the hypergraph, i.e., the difference between the signal components and their respective shifted versions

$$\mathbf{TV}(\mathbf{s}) = \sum_{i=1}^{N} \left| s_i - \sum_{j_1,\ldots,j_{M-1}} F^{\text{norm}}_{ij_1\cdots j_{M-1}} s_{j_1} \cdots s_{j_{M-1}} \right| \tag{42a}$$

$$= \left\| \mathbf{s} - \mathbf{F}^{\text{norm}} \mathbf{s}^{[M-1]} \right\|_1 \tag{42b}$$

where $\mathbf{F}^{\text{norm}} = (1/\lambda_{\max})\mathbf{F}$. Similar to DSP and GSP, pairs of $(\lambda_i, \mathbf{f}_i)$ in (24) characterize the hypergraph spectrum space. A spectrum component with a larger total variation represents a higher frequency component, which indicates faster changes over the hypergraph. Note that as we mentioned in Section III-E, the total variation is larger and the frequency is higher if the corresponding λ is smaller because we usually talk about undirected hypergraph and the λ's are real in the tensor decomposition. To illustrate it more clearly, we consider a hypergraph with nine nodes, five hyperedges, and $m.c.e = 3$ as an example, shown in Fig. 13. As we mentioned before, a smaller λ indicates a higher frequency in HGSP. Hence, we see that the signals have more changes on each vertex if the frequency is higher.

B. Connections to Other Existing Works

We now discuss the relationships between the HGSP and other existing works.

1) Graph Signal Processing: One of the motivations for developing HGSP is to develop a more general framework for signal processing in high-dimensional graphs. Thus, GSP should be a special case of HGSP. We illustrate the GSP-HGSP relationship as follows.

1) *Graphical Models:* GSP is based on the normal graphs [2], where each simple edge connects exactly two nodes; HGSP is based on the hypergraphs, where each hyperedge could connect more than two nodes. Clearly, the normal graph is a special case of hypergraph, for which the *m.c.e* equals two. More specifically, a normal graph is a 2-uniform hypergraph [60]. Hypergraph provides a more general model for multilateral relationships while normal graphs are only able to model bilateral relationships. For example, a 3-uniform hypergraph is able to model the trilateral interaction among users in a social network [61]. As hypergraph is a more general model for high-dimensional interactions, HGSP is also more powerful for high-dimensional signals.
2) *Algebraic Models:* HGSP relies on tensors while GSP relies on matrices, which are second-order tensors. Benefiting from the generality of tensor, HGSP is broadly applicable in high-dimensional data analysis.
3) *Signals and Signal Shifting:* In HGSP, we define the hypergraph signal as $M-1$ times tensor outer product of the original signal. More specifically, the hypergraph signal is the original signal if $M = 2$. Basically, the hypergraph signal is the same as the graph signal if each hyperedge has exactly two nodes. Also shown in Fig. 10(b) of Section III-C, graph shifting is a special case of hypergraph shifting when $M = 2$.
4) *Spectrum Properties:* In HGSP, the spectrum space is defined over the orthogonal-CP decomposition in terms of the basis and coefficients, which is also the E-eigenpairs of the representing tensor [62], shown in (27). In GSP, the spectrum space is defined as the

matrix eigenspace. Since the tensor algebra is an extension of matrix, the HGSP spectrum is also an extension of the GSP spectrum. For example, as discussed in Section III, GFT is the same as HGFT, when $M = 2$.

Overall, HGSP is an extension of GSP, which is both more general and novel. The purpose of developing the HGSP framework is to facilitate more interesting signal processing tasks that involve high-dimensional signal interactions.

2) Higher Order Statistics: Higher order statistics (HOS) has been effectively applied in signal processing [63], [64], which can analyze the multilateral interactions of signal samples and have found successes in many applications, such as blind feature detection [65], decision [66], and signal classifications [67]. In HOS, the kth-order cumulant of random variables $\mathbf{x} = [x_1, \ldots, x_k]^T$ is defined [68] based on the coefficients of $\mathbf{v} = [v_1, \ldots, v_k]^T$ in the Taylor series expansion of cumulant-generating function, i.e.,

$$K(\mathbf{v}) = \ln \mathbf{E}\{\exp(j\mathbf{v}^T\mathbf{x})\}. \tag{43}$$

It is easy to see that HGSP and HOS are related to high-dimensional signal processing. They can be both represented by tensor. For example, in the multichannel problems of [69], the third-order cumulant $\mathbf{C} = \{C_{y_i,y_j,y_z}(t, t_1, t_2)\}$ of zero-mean signals can be represented as a multilinear array, e.g.,

$$C_{y_i,y_j,y_z}(t, t_1, t_2) = \mathbf{E}\{y_i(t)y_j(t + t_1)y_z(t + t_2)\} \tag{44}$$

which is essentially a third-order tensor. More specifically, if there are k samples, the cumulant $\mathbf{C}$ can be represented as an p^k-element vector, which is the flattened signal tensor similar to the n-mode flattening of HGSP signals.

Although both HOS and HGSP are high-dimensional signal processing tools, they focus on complementary aspects of the signals. Specifically, HGSP aims to analyze signals over the high-dimensional vertex domain, while HOS focuses on the statistical domain. In addition, the forms of signal combination are also different, where HGSP signals are based on the hypergraph shifting defined as in (21), whereas HOS cumulants are based on the statistical average of shifted signal products.

3) Learning Over Hypergraphs: Hypergraph learning is another tool to handle the structured data and sometimes uses similar techniques to HGSP. For example, Hein *et al.* [70] proposed an alternative definition of hypergraph total variation and design algorithms in accordance with classification and clustering problems. In addition, hypergraph learning also has its own definition of the hypergraph spectrum space. For example, [39] and [40] represented the hypergraphs using a graph-like similarity matrix and defined a spectrum space as the eigenspace of this similarity matrix. Other works considered different aspects of hypergraph, including the hypergraph Laplacian [71] and hypergraph lifting [21].

The HGSP framework exhibits features different from hypergraph learning.

1) HGSP defines a framework that generalizes the classical digital signal processing and traditional GSP.
2) HGSP applies different definitions of hypergraph characteristics, such as the total variation, spectrum space, and Laplacian.
3) HGSP cares more about the spectrum space while learning focuses more on data.
4) As HGSP is an extension of DSP and GSP, it is more suitable to handle detailed tasks, such as compression, denoising, and detection. All these features make HGSP a different technical concept from hypergraph learning.

V. Tools for Hypergraph Signal Processing

In this section, we introduce several useful tools built within the framework of HGSP.

A. Sampling Theory

Sampling is an important tool in data analysis, which selects a subset of individual data points to estimate the characteristics of the whole population [89]. Sampling plays an important role in applications, such as compression [27] and storage [90]. Similar to sampling signals in time, the HGSP sampling theory can be developed to sample signals over the vertex domain. We now introduce the basics of HGSP sampling theory for lossless signal dimension reduction.

To reduce the size of a hypergraph signal $\mathbf{s}^{[M-1]}$, there are two main approaches: 1) to reduce the dimension of each order and 2) to reduce the number of orders. Since the reduction of order breaks the structure of hypergraph and cannot always guarantee perfect recovery, we adopt the dimension reduction of each order. To change the dimension of a certain order, we can use the n-Mode product. Since each order of the hypergraph signal is equivalent, the n-Mode product operators of each order are the same. Then, the sampling operation of the hypergraph signal is defined as follows.

Definition 11 (Sampling and Interpolation): Suppose that Q is the dimension of each sampled order. The sampling operation is defined as

$$\mathbf{s_Q^{[M-1]}} = \mathbf{s^{[M-1]}} \times_1 \mathbf{U} \times_2 \mathbf{U} \times \cdots \times_{M-1} \mathbf{U} \tag{45}$$

where the sampling operator is $\mathbf{U} \in \mathbb{R}^{Q\times N}$ to be defined later, and the sampled signal is $\mathbf{s_Q^{[M-1]}} \in \mathbb{R}^{\underbrace{Q\times Q\times\cdots\times Q}_{M-1 \text{ times}}}$.

The interpolation operation is defined by

$$\mathbf{s^{[M-1]}} = \mathbf{s_Q^{[M-1]}} \times_1 \mathbf{T} \times_2 \mathbf{T} \times \cdots \times_{M-1} \mathbf{T} \tag{46}$$

where the interpolation operator is $\mathbf{T} \in \mathbb{R}^{N\times Q}$ to be defined later.

As presented in Section III, the hypergraph signal and original signal are different forms of the same data. They may have similar properties in structures. To derive the sampling theory for perfect signal recovery efficiently, we first consider the sampling operations of the original signal.

Definition 12 (Sampling Original Signal): Suppose an original K-bandlimited signal $\mathbf{s} \in \mathbb{R}^N$ is to be sampled into $\mathbf{s_Q} \in \mathbb{R}^Q$, where $q = \{q_1, \ldots, q_Q\}$ denotes the sequence of sampled indices and $q_i \in \{1, 2, \ldots, N\}$. The sampling operator $\mathbf{U} \in \mathbb{R}^{Q\times N}$ is a linearity mapping from $\mathbb{R}^N$ to $\mathbb{R}^Q$, defined by

$$U_{ij} = \begin{cases} 1, & j = q_i \\ 0, & \text{otherwise} \end{cases} \tag{47}$$

and the interpolation operator $\mathbf{T} \in \mathbb{R}^{N\times Q}$ is a linear mapping from $\mathbb{R}^Q$ to $\mathbb{R}^N$. Then, the sampling operation is defined by

$$\mathbf{s_Q} = \mathbf{U} \cdot \mathbf{s} \tag{48}$$

and the interpolation operation is defined by

$$\mathbf{s}' = \mathbf{T} \cdot \mathbf{s_Q}. \tag{49}$$

Analyzing the structure of the sampling operations, we have the following properties.

Theorem 2: The hypergraph signal $\mathbf{s^{[M-1]}}$ shares the same sampling operator $\mathbf{U} \in \mathbb{R}^{Q\times N}$ and interpolation operator $\mathbf{T} \in \mathbb{R}^{N\times Q}$ with the original signal $\mathbf{s}$.

Proof: We first examine one of the orders in n-Mode product of hypergraph signal, i.e., nth-order of $\mathbf{s^{[M-1]}}$, $1 \le n \le N$, as

$$\left(\mathbf{s^{[M-1]}} \times_n \mathbf{U}\right)_{i_1\cdots i_{n-1} j i_{n+1}\cdots i_{M-1}} = \sum_{i_n=1}^{N} s_{i_1} s_{i_2} \cdots s_{i_{M-1}} U_{j i_n}. \tag{50}$$

Since all elements in $\mathbf{s_Q^{[M-1]}}$ should also be the elements of $\mathbf{s^{[M-1]}}$ after sampling, only one $U_{ji_n} = 1$ exists for each j according to (50), i.e., only one term in the summation exists for each j in the right part of (50). Moreover, since $\mathbf{U}$ samples over all the order, $U_{pi_n} = 1$ and $U_{ji_n} = 1$ cannot exist at the same time so that all the entries in $\mathbf{s_Q^{[M-1]}}$ are also in $\mathbf{s^{[M-1]}}$. Suppose $q = \{q_1, q_2, \ldots, q_Q\}$ is the places of nonzero U_{jq_j}'s, we have

$$\mathbf{s_Q^{[M-1]}}(i_1, i_2, \ldots, i_Q) = s_{i_{q_1}} s_{i_{q_2}} \cdots s_{i_{q_Q}}. \tag{51}$$

As a result, we have $U_{ji} = \delta[i - q_j]$, which is the same as the sampling operator for the original signal. For the interpolation operator, the proof is similar and hence omitted. ∎

Given Theorem 2, we only need to analyze the operations of the original signal in the sampling theory. Next, we discuss the conditions for perfect recovery. For the original signal, we have the following property.

Lemma 1: Suppose that $\mathbf{s} \in \mathbb{R}^N$ is a K-bandlimited signal. Then, we have

$$\mathbf{s} = \mathcal{F}_{[K]}^T \tilde{\mathbf{s}}_{[K]} \tag{52}$$

where $\mathcal{F}_{[K]}^T = [\mathbf{f}_1, \ldots, \mathbf{f}_K]$ and $\tilde{\mathbf{s}}_{[K]} \in \mathbb{R}^K$ consists of the first K elements of the original signal in the frequency domain, i.e., $\tilde{\mathbf{s}}$.

Proof: Since $\mathbf{s}$ is K-bandlimited, $\tilde{\mathbf{s}}_i = \mathbf{f}_i^T \mathbf{s} = 0$ when $i > K$. Then, according to (30), we have

$$\begin{aligned}\mathbf{s} = \mathbf{V}^T\mathbf{V}\mathbf{s} &= \sum_{i=1}^{K} \mathbf{f}_i \mathbf{f}_i^T \mathbf{s} + \sum_{i=K+1}^{N} \mathbf{f}_i \mathbf{f}_i^T \mathbf{s} = \sum_{i=1}^{K} \mathbf{f}_i \mathbf{f}_i^T \mathbf{s} + 0 \\ &= \mathcal{F}_{[K]}^T \tilde{\mathbf{s}}_{[K]}\end{aligned} \tag{53}$$

where $\mathbf{V} = [\mathbf{f}_1, \ldots, \mathbf{f}_N]^T$. ∎

This lemma implies that the first K frequency components carry all the information of the original signal. Since the hypergraph signal and the original signal share the same sampling operators, we can reach a similar conclusion for perfect recovery as [27] and [28], given in the following theorem.

Theorem 3: Define the sampling operator $\mathbf{U} \in \mathbb{R}^{Q\times N}$ according to $U_{ji} = \delta[i - q_j]$ where $1 \le q_i \le N$, $i = 1, \ldots, Q$. By choosing $Q \ge K$ and the interpolation operator $\mathbf{T} = \mathcal{F}_{[K]}^T \mathbf{Z} \in \mathbb{R}^{N\times Q}$ with $\mathbf{U}\mathcal{F}_{[K]}^T = \mathbf{I_K}$ and $\mathcal{F}_{[K]}^T = [\mathbf{f}_1, \ldots, \mathbf{f}_K]$, we can achieve a perfect recovery, i.e., $\mathbf{s} = \mathbf{TUs}$ for all K-bandlimited original signal $\mathbf{s}$ and the corresponding hypergraph signal $\mathbf{s}^{[M-1]}$.

Proof: To prove the theorem, we show that $\mathbf{TU}$ is a projection operator and $\mathbf{T}$ spans the space of the first K eigenvectors. From Lemma 1 and $\mathbf{s} = \mathbf{Ts_Q}$, we have

$$\mathbf{s} = \mathcal{F}_{[K]}^T \tilde{\mathbf{s}}_{[K]} = \mathcal{F}_{[K]}^T \mathbf{Z}\mathbf{s_Q}. \tag{54}$$

As a result, $\operatorname{rank}(\mathbf{Zs_Q}) = \operatorname{rank}(\tilde{\mathbf{s}}_{[K]}) = K$. Hence, we conclude that $K \le Q$.

Next, we show that $\mathbf{TU}$ is a projection by proving that $\mathbf{TU} \cdot \mathbf{TU} = \mathbf{TU}$. Since, we have $Q \ge K$ and

$$\mathbf{ZU}\mathcal{F}_{[K]}^T = \mathbf{I_K}. \tag{55}$$

We have

$$\begin{aligned}\mathbf{TU} \cdot \mathbf{TU} &= \mathcal{F}_{[K]}^T \mathbf{ZU} \mathcal{F}_{[K]}^T \mathbf{ZU} && \text{(56a)}\\ &= \mathcal{F}_{[K]}^T \mathbf{ZU} = \mathbf{TU}. && \text{(56b)}\end{aligned}$$

Hence, TU is a projection operator. For the spanning part, the proof is the same as that in [27]. ∎

Theorem 3 shows that a perfect recovery is possible for a bandlimited hypergraph signal. We now examine some interesting properties of the sampled signal.

From the previous discussion, we have $\tilde{\mathbf{s}}_{[K]} = \mathbf{Zs_Q}$, which has a similar form to HGFT, where $\mathbf{Z}$ can be treated as the Fourier transform operator. Suppose that $Q = K$ and $\mathbf{Z} = [\mathbf{z}_1 \quad \cdots \quad \mathbf{z}_K]^T$. We have the following first-order difference property.

Theorem 4: Define a new hypergraph by $\mathbf{F_K} = \sum_{i=1}^{K} \lambda_i \cdot \mathbf{z}_i \circ \cdots \circ \mathbf{z}_i$. Then, for all K-bandlimited signal $\mathbf{s}^{[M-1]} \in \mathbb{R}^{\underbrace{N\times N\times\cdots\times N}_{M \text{ times}}}$, it holds that

$$\mathbf{s}_{[K]} - \mathbf{F_K}\mathbf{s}_{[K]}^{[M-1]} = \mathbf{U}\left(\mathbf{s} - \mathbf{F}\mathbf{s}^{[M-1]}\right). \tag{57}$$

Proof: Let the diagonal matrix $\Sigma_{[K]}$ consist of the first K coefficients $\{\lambda_1, \ldots, \lambda_K\}$. Since $\mathbf{U}\mathcal{F}_{[K]}^T = \mathbf{I_K}$, we have

$$\begin{aligned}\mathbf{F_K}\mathbf{s}_{[K]}^{[M-1]} &= \mathbf{Z}^{-1}\Sigma_{[K]}\hat{\mathbf{s}}_{[K]} && \text{(58a)}\\ &= \mathbf{U}\mathcal{F}_{[K]}^T\Sigma_{[K]}\hat{\mathbf{s}}_{[K]} && \text{(58b)}\\ &= \mathbf{U}\left[\left(\sum_{i=1}^{K}\lambda_i \cdot \underbrace{\mathbf{f}_i \circ \cdots \circ \mathbf{f}_i}_{M \text{ times}}\right)\left(\underbrace{\mathbf{s}\circ\cdots\circ\mathbf{s}}_{M-1 \text{ times}}\right) + 0\right] && \text{(58c)}\\ &= \mathbf{U}\mathbf{F}\mathbf{s}^{[M-1]}. && \text{(58d)}\end{aligned}$$

Since $\mathbf{s}_{[K]} = \mathbf{Us}$, it therefore holds that $\mathbf{s}_{[K]} - \mathbf{F_K}\mathbf{s}_{[K]}^{[M-1]} = \mathbf{U}(\mathbf{s} - \mathbf{F}\mathbf{s}^{[M-1]})$. ∎

Theorem 4 shows that the sampled signals form a new hypergraph that preserves the information of the one-time shifting filter over the original hypergraph. For example, the left-hand side of (57) represents the difference between the sampled signal and the one-time shifted version in the new

hypergraph. The right-hand side of (57) is the difference between a signal and its one-time shifted version in the original hypergraph, together with the sampling operator. That is, the sampled result of the one-time shifting differences in the original hypergraph is equal to the one-time shifting differences in the new sampled hypergraph.

B. Filter Design

Filter is an important tool in signal processing applications, such as denoising, feature enhancement, smoothing, and classification. In GSP, the basic filtering is defined as $\mathbf{s}' = \mathbf{F_M s}$ where $\mathbf{F_M}$ is the representing matrix [2]. In HGSP, the basic hypergraph filtering is defined in Section III-C as $\mathbf{s}_{(1)} = \mathbf{F}\mathbf{s}^{[M-1]}$, which is designed according to the tensor contraction. The HGSP filter is a multilinear mapping [72]. The high-dimensionality of tensors provides more flexibility in designing the HGSP filter.

1) Polynomial Filter Based on Representing Tensor: Polynomial filter is one basic form of HGSP filters, with which signals are shifted several times over the hypergraph. An example of polynomial filter is given as Fig. 8 in Section III-B. A k-time shifting filter is defined as

$$\mathbf{s}_{(k)} = \mathbf{F}\mathbf{s}_{(k-1)}^{[M-1]} \tag{59a}$$

$$= \underbrace{\mathbf{F}\left(\mathbf{F}\left(\cdots\left(\mathbf{F}\mathbf{s}^{[M-1]}\right)^{[M-1]}\right)^{[M-1]}\right)^{[M-1]}}_{k \text{ times}}. \tag{59b}$$

More generally, a polynomial filter is designed as

$$\mathbf{s}' = \sum_{k=1}^{a} \alpha_k \mathbf{s}_{(k)} \tag{60}$$

where $\{\alpha_k\}$ are the filter coefficients. Such HGSP filters are based on the multilinear tensor contraction, which could be used for different signal processing tasks by selecting the specific parameters a and $\{\alpha_i\}$.

In addition to the general polynomial filter based on hypergraph signals, we provide another specific form of polynomial filter based on the original signals. As mentioned in Section III-E, the supporting matrix $\mathbf{P_s}$ in (33) captures all the information of the frequency space. For example, the un-normalized supporting matrix $\mathbf{P} = \lambda_{\max}\mathbf{P_s}$ is calculated as

$$\mathbf{P} = \begin{bmatrix}\mathbf{f}_1 & \cdots & \mathbf{f}_N\end{bmatrix}\begin{bmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_N\end{bmatrix}\begin{bmatrix}\mathbf{f}_1^T \\ \vdots \\ \mathbf{f}_N^T\end{bmatrix}. \tag{61}$$

Obviously, the hypergraph spectrum pair $(\lambda_r, \mathbf{f}_r)$ is an eigenpair of the supporting matrix $\mathbf{P}$. Moreover, Theorem 1 shows that the total variation of frequency component equals to a function of $\mathbf{P}$, i.e.,

$$\mathbf{TV}(\mathbf{f}_r) = \left\|\mathbf{f}_r - \frac{1}{\lambda_{\max}}\mathbf{P}\mathbf{f}_r\right\|_1. \tag{62}$$

From (62), $\mathbf{P}$ can be interpreted as a shifting matrix for the original signal. Accordingly, we can design a polynomial filter for the original signal based on the supporting matrix $\mathbf{P}$ whose kth-order term is defined as

$$\mathbf{s}_{<k>} = \mathbf{P}^k\mathbf{s}. \tag{63}$$

The ath-order polynomial filter is simply given as

$$\mathbf{s}' = \sum_{k=1}^{a} \alpha_k \mathbf{P}^k\mathbf{s}. \tag{64}$$

A polynomial filter over the original signal can be determined with specific choices of a and α.

Let us consider some interesting properties of the polynomial filter for the original signal. First, given the kth-order term, we have the following property as Lemma 2.

Lemma 2:

$$\mathbf{s}_{<k>} = \sum_{r=1}^{N} \lambda_r^k\left(\mathbf{f}_r^T\mathbf{s}\right)\mathbf{f}_r. \tag{65}$$

Proof: Let $\mathbf{V}^T = [\mathbf{f}_1, \ldots, \mathbf{f}_N]$ and $\Sigma = diag([\lambda_1, \ldots, \lambda_N])$. Since $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, we have

$$\mathbf{P}^k = \underbrace{\mathbf{V}^T\Sigma\mathbf{V}\mathbf{V}^T\Sigma\mathbf{V}\cdots\mathbf{V}^T\Sigma\mathbf{V}}_{k \text{ times}} \tag{66a}$$

$$= \mathbf{V}^T\Sigma^k\mathbf{V}. \tag{66b}$$

Therefore, the kth-order term is given as

$$\mathbf{s}_{<k>} = \mathbf{V}^T\Sigma^k\mathbf{V}\mathbf{s} \tag{67a}$$

$$= \sum_{r=1}^{N} \lambda_r^k\left(\mathbf{f}_r^T\mathbf{s}\right)\mathbf{f}_r. \tag{67b}$$

■

From Lemma 2, we obtain the following property of the polynomial filter for the original signal.

Theorem 5: Let $h(\cdot)$ be a polynomial function. For the polynomial filter $\mathbf{H} = h(\mathbf{P})$ for the original signal, the filtered signal satisfies

$$\mathbf{H}\mathbf{s} = \sum_{r=1}^{N} h(\mathbf{s}_{<i>}) = \sum_{r=1}^{N} h(\lambda_r)\mathbf{f}_r\left(\mathbf{f}_r^T\mathbf{s}\right). \tag{68}$$

This theorem works as the invariance property of exponential in HGSP, similar to those in GSP and DSP [2]. Equations (60) and (63) provide more choices for HGSP polynomial filters in HGSP and data analysis. We will give specific examples of practical applications in Section VI.

2) General Filter Design Based on Optimization: In GSP, some filters are designed via optimization formulations [2], [73], [74]. Similarly, general HGSP filters can also be designed via optimization approaches. Assume $\mathbf{y}$ is the observed signal before shifting and $\mathbf{s} = h(\mathbf{F}, \mathbf{y})$ is the shifted signal by HGSP filter $h(\cdot)$ designed for specific applications. Then, the filter design can be formulated as

$$\min_h \|\mathbf{s} - \mathbf{y}\|_2^2 + \gamma f(\mathbf{F}, \mathbf{s}) \tag{69}$$

where $\mathbf{F}$ is the representing tensor of the hypergraph and $f(\cdot)$ is a penalty function designed for specific problems. For example, the total variation could be used as a penalty function for the purpose of smoothness. Other alternative penalty functions include the label rank, Laplacian regularization, and spectrum. In Section VI, we shall provide some filter design examples.

TABLE I
COMPRESSION RATION OF DIFFERENT METHODS

size	16×16							256×256			
image	Radiation	People	load	inyang	stop	error	smile	lenna	mri	ct	AVG
IANH-HGSP	**1.52**	**1.45**	**1.42**	**1.47**	**1.52**	**1.39**	**1.40**	**1.57**	**1.53**	**1.41**	**1.47**
(α, β)-GSP	1.37	1.23	1.10	1.26	1.14	1.16	1.28	1.07	1.11	1.07	1.18
4 connected-GSP	1.01	1.02	1.01	1.01	1.04	1.02	1.07	1.04	1.05	1.07	1.03

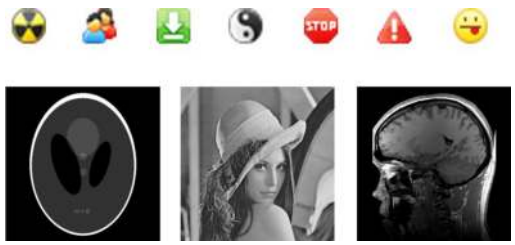

Fig. 14. Test set of images.

VI. APPLICATION EXAMPLES

In this section, we consider several application examples for our newly proposed HGSP framework. These examples illustrate the practical use of HGSP in some traditional tasks, such as filter design and efficient data representation. We also consider problems in data analysis, such as classification and clustering.

A. Data Compression

Efficient representation of signals is important in data analysis and signal processing. Among many applications, data compression attracts significant interests for efficient storage and transmission [75]–[77]. Projecting signals into a suitable orthonormal basis is a widely used compression method [5]. Within the proposed HGSP framework, we propose a data compression method based on the HGFT. We can represent N signals in the original domain with C frequency coefficients in the hypergraph spectrum domain. More specifically, with the help of the sampling theory in Section V, we can compress an K-bandlimited signal of N signal points losslessly with K spectrum coefficients.

To test the performance of our HGSP compression and demonstrate that hypergraphs may be a better representation of structured signals than normal graphs, we compare the results of image compression with those from the GSP-based compression method [5]. We test over seven small-size 16×16 icon images and three-size 256×256 photograph images, shown in Fig. 14.

The HGSP-based image compression method is described as follows. Given an image, we first model it as a hypergraph with the image adaptive neighborhood hypergraph (IANH) model [30]. To reduce complexity, we pick three closest neighbors in each hyperedge to construct a third-order adjacency tensor. Next, we can calculate the Fourier basis of the adjacency tensor as well as the bandwidth K of the hypergraph signals. Finally, we can represent the original images using C spectrum coefficients with $C = K$. For a large image, we may first cut it into smaller image blocks before applying HGSP compression to improve speed.

For the GSP-based method in [5], we represent the images as graphs with: 1) the 4-connected neighbor model [31] and 2) the distance-based model in which an edge exists only if the spatial distance is below α and the pixel distance is below β. The graph Fourier space and corresponding coefficients in the frequency domain are then calculated to represent the original image.

We use the compression ratio $\text{CR} = N/C$ to measure the efficiency of different compression methods. A large CR implies higher compression efficiency. The result is summarized in Table I, from which we can see that our HGSP-based compression method achieves higher efficiency than the GSP-based compression methods.

In addition to the image data sets, we also test the efficiency of HGSP spectrum compression over the MovieLens data set [83], where each movie data point has rating scores and tags from viewers. Here, we treat scores of movies as signals and construct graph models based on the tag relationships. Similar to the game data set shown in Fig. 2(b), two movies are connected in a normal graph if they have similar tags. For example, if movies are labeled with "love" by users, they are connected by an edge. To model the data set as a hypergraph, we include the movies into one hyperedge if they have similar tags. For convenience and complexity, we set $m.c.e = 3$. With the graph and hypergraph models, we compress the signals using the sampling method discussed earlier. For lossless compression, our HGSP method is able to use only 11.5% of the samples from the original signals to recover the original data set by choosing the suitable additional basis (see Section III-F). On the other hand, the GSP method requires 98.6% of the samples. We also test the error between the recovered and original signals based on the varying numbers of samples. As shown in Fig. 15, the recovery error naturally decreases with more samples. Note that our HGSP method achieves a much better performance once it obtains sufficient number of samples, while GSP error drops slowly. This is due to the first few key HGSP spectrum basis elements carry most of the original information, thereby leading to a more efficient representation for structured data sets.

Overall, hypergraph and HGSP lead to more efficient descriptions of the structured data in most applications. With a more suitable hypergraph model and more developed methods, the HGSP framework could be a very new important tool in data compression.

B. Spectral Clustering

Clustering problem is widely used in a variety of applications, such as social network analysis, computer vision, and communication problems. Among many methods, spectral

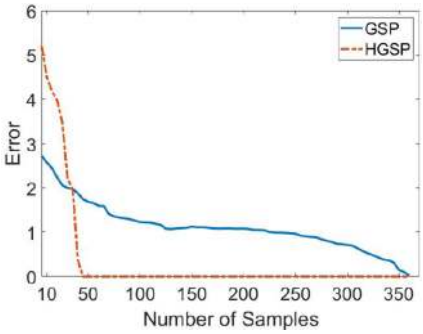

Fig. 15. Errors of compression methods over the MovieLen Data set: y-axis shows the error between the original signals and recovered signals, and x-axis is the number of samples.

Algorithm 1 HGSP Fourier Spectral Clustering

1: **Input**: Dataset modeled in hypergraph $\mathcal{H}$, the number of clusters k.
2: Construct adjacency tensor $\mathbf{A}$ in Eq. (15) from the hypergraph $\mathcal{H}$.
3: Apply orthogonal decomposition to $\mathbf{A}$ and compute Fourier basis $\mathbf{f_i}$ together with Fourier frequency coefficient λ_i using Eq. (24).
4: Find the first E leading Fourier basis $\mathbf{f}_i$ with $\lambda_i \neq 0$ and construct a Fourier spectrum matrix $\mathbf{S} \in \mathbb{R}^{N\times E}$ with columns as the leading Fourier basis.
5: Cluster the rows of $\mathbf{S}$ into k clusters using k-means clustering.
6: Put node i in partition j if the ith row is assigned to the jth cluster.
7: **Output**: k partitions of the hypergraph data set.

clustering is an efficient clustering method [6], [37]. Modeling the data set by a normal graph before clustering the data spectrally, significant improvement is possible in structured data [91]. However, such standard spectral clustering methods only exploit pairwise interactions. For applications where the interactions involve more than two nodes, hypergraph spectral clustering should be a more natural choice.

In hypergraph spectral clustering, one of the most important issues is how to define a suitable spectral space. Li *et al.* [39] and Ahn *et al.* [40] introduced the hypergraph similarity spectrum for spectral clustering. Before spectral clustering, they first modeled the hypergraph structure into a graph-like similarity matrix. They then defined the hypergraph spectrum based on the eigenspace of the similarity matrix. However, since the modeling of hypergraph with a similarity matrix may result in certain loss of the inherent information, a more efficient spectral space defined directly over hypergraph is more desired as introduced in our HGSP framework. With HGSP, as the hypergraph Fourier space from the adjacency tensor has a similar form to the spectral space from adjacency matrix in GSP, we could develop the spectral clustering method based on the hypergraph Fourier space as in Algorithm 1.

To test the performance of the HGSP spectral clustering, we compare the achieved results with those from the hypergraph similarity method (HSC) in [40], using the zoo data set [34].

Algorithm 2 LP-HGSP Classification

1: **Input**: Dataset $\mathbf{s}$ consisting of labeled training data and unlabeled test data.
2: Establish a hypergraph by similarities and set unlabeled data as $\{0\}$ in the signal $\mathbf{s}$.
3: Train the coefficients α of the LP-HGSP filter in Eq. (70) by minimizing the errors of signs of training data in $\mathbf{s}' = \mathbf{Hs}$.
4: Implement LP-HGSP filter. If $\mathbf{s}'_i > 0$, the ith data is labeled as 1; otherwise, it is labeled as -1.
5: **Output**: Labels of test signals.

To measure the performance, we compute the intracluster variance and the average Silhouette of nodes [41]. Since we expect the data points in the same cluster to be closer to each other, the performance is considered better if the intracluster variance is smaller. On the other hand, the Silhouette value is a measure of how similar an object is to its own cluster versus other clusters. A higher Silhouette value means that the clustering configuration is more appropriate.

The comparative results are shown in Fig. 16. Form the test result, we can see that our HGSP method generates a lower variance and a higher Silhouette value. More intuitively, we plot the clusters of animals in Fig. 17. Cluster 2 covers small animals like bugs and snakes. Cluster 3 covers carnivores whereas Cluster 7 groups herbivores. Cluster 4 covers birds and Cluster 6 covers fish. Cluster 5 contains the rodents such as mice. One interesting category is Cluster 1: although dolphins, sea-lions, and seals live in the sea, they are mammals and are clustered separately from Cluster 6. From these results, we see that the HGSP spectral clustering method could achieve better performance and our definition of hypergraph spectrum may be more appropriate for spectral clustering in practice.

C. Classification

Classification problems are important in data analysis. Traditionally, these problems are studied by the learning methods [35]. Here, we propose an HGSP-based method to solve the $\{\pm 1\}$ classification problem, where a hypergraph filter serves as a classifier.

The basic idea adopted for the classification filter design is label propagation (LP), where the main steps are to first construct a transmission matrix and then propagate the label based on the transmission matrix [36]. The label will converge after a sufficient number of shifting steps. Let $\mathbf{W}$ be the propagation matrix. Then the label could be determined by the distribution $\mathbf{s}' = \mathbf{W}^k\mathbf{s}$. We see that $\mathbf{s}'$ is in the form of filtered graph signal. Recall that in Section V-B, the supporting matrix $\mathbf{P}$ has been shown to capture the properties of hypergraph shifting and total variation. Here, we propose an HGSP classifier based on the supporting matrix $\mathbf{P}$ defined in (61) to generate matrix

$$\mathbf{H} = (\mathbf{I} + \alpha_1\mathbf{P})(\mathbf{I} + \alpha_2\mathbf{P}) \cdots (\mathbf{I} + \alpha_k\mathbf{P}). \tag{70}$$

Our HGSP classifier is to simply rely on sign[$\mathbf{Hs}$]. The main steps of the propagated LP-HGSP classification method is described in Algorithm 2.

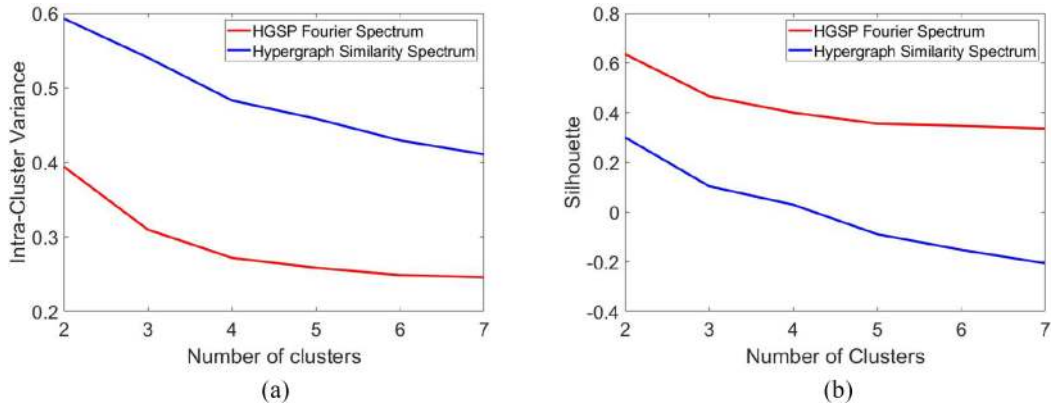

Fig. 16. Performance of hypergraph spectral clustering. (a) Variance in the same cluster. (b) Average Silhouette in the hypergraph.

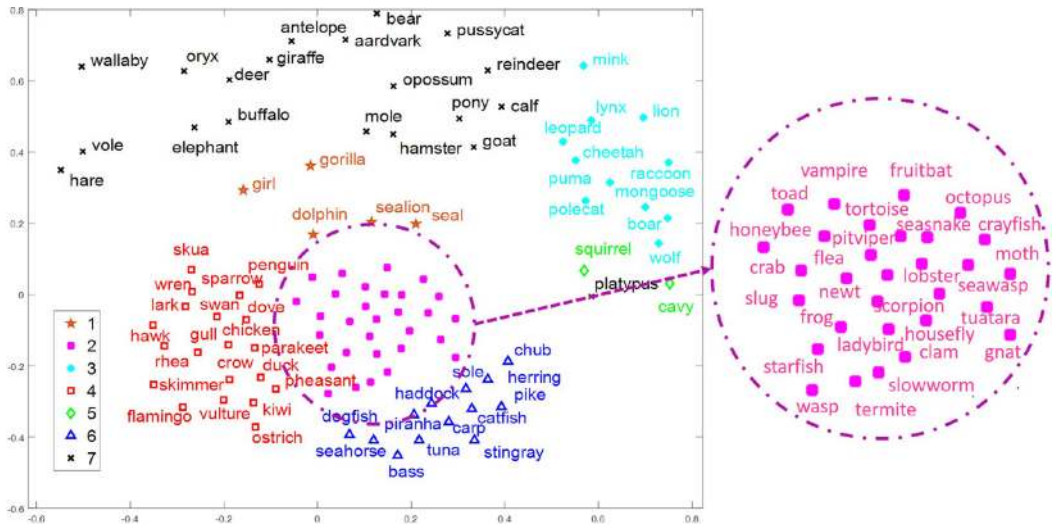

Fig. 17. Cluster of animals.

To test the performance of the hypergraph-based classifier, we implement them over the zoo data sets. We determine whether the animals have hair based on other features, formulated as a $\{\pm 1\}$ classification problem. We randomly pick different percentages of training data and leave the remaining data as the test set among the total 101 data points. We smooth the curve with 1000 combinations of randomly picked training sets. We compare the HGSP-based method against the SVM method with the RBF kernel and the LP-GSP method [2]. In the experiment, we model the data set as hypergraph or graph based on the distance of data. The threshold of determining the existence of edges is designed to ensure the absence of isolated nodes in the graph. For the LP method, we set $k = 15$. The result is shown in Fig. 18(a). From the result, we see that the LP HGSP method (LP-HGSP) is moderately better than LP-GSP. The graph-based methods, i.e., LP-GSP and LP-HGSP, both performs better than SVM. The performance of SVM appears less satisfactory, likely because the data set is rather small. The model-based graph and hypergraph methods are rather robust when applied to such small data sets. To illustrate this effect more clearly, we tested the SVM and hypergraph performance with new configurations by the increasing data set size and the fixing ratio of training data in Fig. 18(b). In the experiment, we first pick different sizes of data subsets from the original zoo data set randomly as the new data sets. Then, with each size of the new data set, 40% data points are randomly picked as the training data, and the remaining data points are used as the test data. We average the results over 10 000 times of experiments to smooth the curve. We can see from Fig. 18(b) that the performance of SVM shows significant improvement as the data set size grows larger. This comparison indicates that SVM may require more data to achieve better performance, as shown in the comparative results of Fig. 18(a). Generally, the HGSP-based method exhibits better overall performance and shows significant advantages with small data sets. Although GSP and HGSP classifiers are both model-based, hypergraph-based ones usually perform better than graph-based ones, since hypergraphs provide a better description of the structured data in most applications.

D. Denoising

Signals collected in the real world often contain noises. Signal denoising is thus an important application in signal processing. Here, we design a hypergraph filter to implement signal denoising.

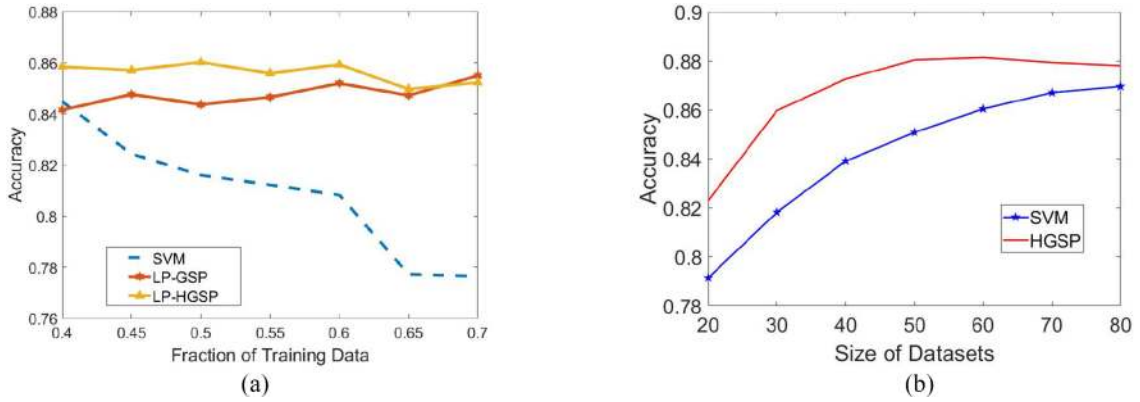

Fig. 18. Performance of classifiers. (a) Over data sets with a fixed datasize and different ratios of training data. (b) Over data sets with different datasizes and a fixed ratio of training data.

TABLE II
MSE OF FILTERED SIGNAL

γ	10e-5	10e-4	10e-3	10e-2	10e-1	1	10
Uniform Distribution: U(0, 0.1)							
GSP	0.0031	0.0031	0.0031	0.0026	**0.0017**	0.0895	0.4523
HGSP	0.0031	0.0031	0.0028	**0.0012**	0.0631	0.1876	0.4083
Wiener	0.0201						
Median	0.0142						
Normal Distribution: N(0, 0.09)							
GSP	0.790	0.790	0.0786	**0.0556**	0.0604	0.1286	0.4681
HGSP	0.0790	0.0585	**0.0305**	0.0778	0.1235	0.2374	0.4176
Wiener	0.0368						
Median	0.0359						
Normal Distribution: N(-0.02, 0.0001)							
GSP	**5.34e-04**	5.36e-04	5.54e-04	7.76e-04	0.0055	0.1113	0.4650
HGSP	**4.17e-04**	4.72e-04	4.86e-04	6.48e-04	0.0044	0.0868	0.3483
Wiener	0.0230						
Median	0.0096						

As mentioned in Section III, the smoothness of a graph signal, which describes the variance of hypergraph signals, could be measured by the total variation. Assume that the original signal is smooth. We formulate signal denoising as an optimization problem. Suppose that $\mathbf{y} = \mathbf{s} + \mathbf{n}$ is a noisy signal with noise $\mathbf{n}$, and $\mathbf{s}' = h(\mathbf{F}, \mathbf{y})$ is the denoised data by the HGSP filter $h(\cdot)$. The denoising problem could be formulated as an optimization problem

$$\min_h \left\|\mathbf{s}' - \mathbf{y}\right\|_2^2 + \gamma \cdot \left\|\mathbf{s}' - \mathbf{s}'^{\text{norm}}_{<1>}\right\|_2^2 \tag{71}$$

where the second term is the weighted quadratic total variation of the filtered signal $\mathbf{s}'$ based on the supporting matrix.

The denoising problem of (71) aims to smooth the signal based on the original noisy data $\mathbf{y}$. The first term keeps the denoised signal close to the original noisy signal, whereas the second term tries to smooth the recovered signal. Clearly, the optimized solution of filter design is

$$\mathbf{s}' = h(\mathbf{F}, \mathbf{y}) = [\mathbf{I} + \gamma(\mathbf{I} - \mathbf{P_s})^{\mathbf{T}}(\mathbf{I} - \mathbf{P_s})]^{-1}\mathbf{y} \tag{72}$$

where $\mathbf{P}_s = \sum_{i=1}^{N}(\lambda_i/\lambda_{\max})\mathbf{f}_i\mathbf{f}_i^{\mathbf{T}}$ describes a hypergraph Fourier decomposition. From (72), we see that the solution is in the form of $\mathbf{s}' = \mathbf{H}\mathbf{y}$ for denoising, which adopts a hypergraph filter $h(\cdot)$ as

$$\mathbf{H} = \left[\mathbf{I} + \gamma(\mathbf{I} - \mathbf{P_s})^{\mathbf{T}}(\mathbf{I} - \mathbf{P_s})\right]^{-1}. \tag{73}$$

The HGSP-based filter follows a similar idea to GSP-based denoising filter [33]. However, different definitions of the total variation and signal shifting result in the different designs of HGSP versus GSP filters. To test the performance, we compare our method with the basic Wiener filter, Median filter, and GSP-based filter [33] using the image data sets of Fig. 14. We apply different types of noises. To quantify the filter performance, we use the mean square error (MSE) between each true signal and the corresponding signal after filtering. The results are given in Table II. From these results, we can see that for each type of noise and picking optimized γ for all the methods, our HGSP-based filter out-performs other filters.

E. Other Potential Applications

In addition to the application algorithms discussed above, there could be many other potential applications for HGSP. In this section, we suggest several potential applicable data sets and systems for HGSP.

1) *IoT:* With the development of IoT techniques, the system structures become increasingly complex, which makes traditional graph-based tools inefficient to handle the high-dimensional interactions. On the other hand, the hypergraph-based HGSP is powerful in dealing with high-dimensional analysis in the IoT system: for example, data intelligence over sensor networks, where

hypergraph-based analysis has already attracted significant attentions [92], and HGSP could be used to handle tasks like clustering, classification, and sampling.

2) *Social Network:* Another promising application is the analysis of social network data sets. As discussed earlier, a hyperedge is an efficient representation for the multilateral relationship in social networks [15], [80]; HGSP can then be effective in analyzing multilateral node interactions.
3) *Natural Language Processing:* Furthermore, natural language processing is an area that can benefit from HGSP. Modeling the sentence and language by hypergraphs [81], [82], HGSP can be a tool for language classification and clustering tasks.

Overall, due to its systematic and structural approach, HGSP is expected to become an important tool in handling high-dimensional signal processing tasks that are traditionally addressed by DSP- or GSP-based methods.

VII. Conclusion

In this article, we proposed a novel tensor-based framework of HGSP that generalizes the traditional GSP to high-dimensional hypergraphs. This article provided important definitions in HGSP, including hypergraph signals, hypergraph shifting, HGSP filters, frequency, and bandlimited signals. We presented basic HGSP concepts, such as the sampling theory and filtering design. We show that hypergraph can serve as an efficient model for many complex data sets. We also illustrate multiple practical applications for HGSP in signal processing and data analysis, where we provided numerical results to validate the advantages and the practicality of the proposed HGSP framework. All the features of HGSP make it a powerful tool for IoT applications in the future.

Future Directions: With the development of tensor algebra and hypergraph spectra, more opportunities are emerging to explore HGSP and its applications. One interesting topic is how to construct the hypergraph efficiently, where distance-based and model-based methods have achieved significant successes in specific areas, such as image processing [93] and natural language processing [94]. Another promising direction is to apply HGSP in analyzing and optimizing the multilayer networks. As we discussed in the introduction, hypergraph is an alternative model to present the multilayer network [13], and HGSP becomes a useful tool when dealing with the multilayer structures. Other future directions include the development of fast operations, such as the fast HGFT and applications over high-dimensional data sets [95].

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, Apr. 2013.

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[3] M. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 1, pp. 2566–2572, Feb. 2002.

[4] S. Barbarossa and M. Tsitsvero, "An introduction to hypergraph signal processing," in *Proc. Acoust. Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 6425–6429.

[5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Apr. 2014.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[7] R. Wagner, V. Delouille, and R. G. Baraniuk, "Distributed wavelet denoising for sensor networks," in *Proc. 45th IEEE Conf. Decis. Control*, San Diego, CA, USA, Dec. 2006, pp. 373–379.

[8] S. K. Narang and A. Ortega, "Local two-channel critically sampled filter-banks on graphs," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Hong Kong, Sep. 2010, pp. 333–336.

[9] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *Proc. ICASSP*, Kyoto, Japan, Mar. 2012, pp. 3921–3924.

[10] S. Zhang, H. Zhang, H. Li, and S. Cui, "Tensor-based spectral analysis of cascading failures over multilayer complex systems," in *Proc. 56th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, Oct. 2018, pp. 997–1004.

[11] Z. Huang, C. Wang, M. Stojmenovic, and A. Nayak, "Characterization of cascading failures in interdependent cyberphysical systems," *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2158–2168, Aug. 2015.

[12] T. Michoel and B. Nachtergaele, "Alignment and integration of complex networks by hypergraph-based spectral clustering," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 86, no. 5, Nov. 2012, Art. no. 056111.

[13] B. Oselio, A. Kulesza, and A. Hero, "Information extraction from large multi-layer social networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Brisbane, QLD, Australia, Apr. 2015, pp. 5451–5455.

[14] M. De Domenico *et al.*, "Mathematical formulation of multilayer networks," *Phys. Rev. X*, vol. 3, no. 4, Dec. 2013, Art. no. 041022.

[15] G. Ghoshal, V. Zlatić, G. Caldarelli, and M. E. J. Newman, "Random hypergraphs and their applications," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 79, no. 6, Jun. 2009, Art. no. 066118.

[16] L. J. Grady and J. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. London, U.K.: Springer, 2010.

[17] C. G. Drake *et al.*, "Analysis of the New Zealand black contribution to lupus-like renal disease. Multiple genes that operate in a threshold manner," *J. Immunol.*, vol. 154, no. 5, pp. 2441–2447, Mar. 1995.

[18] V. I. Voloshin, *Introduction to Graph and Hypergraph Theory*. New York, NY, USA: Nova Sci., 2009.

[19] C. Berge and E. Minieka, *Graphs and Hypergraphs*. Amsterdam, The Netherlands: North-Holland, 1973.

[20] A. Banerjee, C. Arnab, and M. Bibhash, "Spectra of general hypergraphs," *Linear Algebra Appl.*, vol. 518, pp. 14–30, Dec. 2016.

[21] S. Kok and P. M. Domingos, "Learning Markov logic network structure via hypergraph lifting," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 505–512.

[22] N. Yadati, M. Nimishakavi, P. Yadav, A. Louis, and P. Talukdar, "HyperGCN: Hypergraph convolutional networks for semi-supervised classification," *arXiv:1809.02589 [cs]*, Sep. 2018.

[23] A. Mithani, G. M. Preston, and J. Hein, "Rahnuma: Hypergraph-based tool for metabolic pathway prediction and network comparison," *Bioinformatics*, vol. 25, no. 14, pp. 1831–1832, Jul. 2009.

[24] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Aug. 2009.

[25] L. Qi, "Eigenvalues of a real supersymmetric tensor," *J. Symbolic Comput.*, vol. 40, no. 6, pp. 1302–1324, Jun. 2005.

[26] A. Afshar *et al.*, "CP-ORTHO: An orthogonal tensor factorization framework for spatio-temporal data," in *Proc. 25th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Redondo Beach, CA, USA, Jan. 2017, p. 67.

[27] S. Chen, A. Sandryhaila, and J. Kovaćević, "Sampling theory for graph signals," in *Proc. Acoust. Speech Signal Process. (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015, pp. 3392–3396.

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

[29] B. W. Bader and T. G. Kolda. (Feb. 2015). *MATLAB Tensor Toolbox Version 2.6*. [Online]. Available: http://www.sandia.gov/∼tgkolda/TensorToolbox/

[30] A. Bretto and L. Gillibert, "Hypergraph-based image representation," in *Proc. Int. Workshop Graph Based Represent. Pattern Recognit.*, Berlin, Germany, Apr. 2005, pp. 1–11.

[31] A. Morar, F. Moldoveanu, and E. Gröller, "Image segmentation based on active contours without edges," in *Proc. IEEE 8th Int. Conf. Intell. Comput. Commun. Process.*, Cluj-Napoca, Romania, Sep. 2012, pp. 213–220.

[32] K. Fukunaga, S. Yamada, H. S. Stone, and T. Kasai, "A representation of hypergraphs in the Euclidean space," *IEEE Trans. Comput.*, vol. C-33, no. 4, pp. 364–367, Apr. 1984.

[33] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal denoising on graphs via graph filtering," in *Proc. Signal Inf. Process. (GlobalSIP)*, Atlanta, GA, USA, Dec. 2014, pp. 872–876.

[34] D. Dua and E. K. Taniskidou. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[35] D. L. Medin and M. Schaffer, "Context theory of classification learning," *Psychol. Rev.*, vol. 85, no. 3, pp. 207–238, 1978.

[36] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," School Comput. Sci., Carnegie Mellon Univ., Rep. CMU-CALD-02–107, 2002.

[37] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. Adv. Neural Inf. Process. Syst.*. Sep. 2007, pp. 1601–1608.

[38] E. O. Brigham and E. O. Brigham, *The Fast Fourier Transform and Its Applications*. Englewood Cliffs, NJ, USA: Prentice Hall, 1988.

[39] X. Li, W. Hu, C. Shen, A. Dick, and Z. Zhang, "Context-aware hypergraph construction for robust spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2588–2597, Oct. 2014.

[40] K. Ahn, K. Lee, and C. Suh, "Hypergraph spectral clustering in the weighted stochastic block model," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 5, pp. 959–974, Oct. 2018.

[41] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[42] G. Li, L. Qi, and G. Yu, "The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory," *Numer. Linear Algebra Appl.*, vol. 20, no. 6, pp. 1001–1029, Mar. 2013.

[43] R. B. Bapat, *Graphs and Matrices*. London, U.K.: Springer, 2010.

[44] K. C. Chang, K. Pearson, and T. Zhang, "On eigenvalue problems of real symmetric tensors," *J. Math. Anal. Appl.*, vol. 305, no. 1, pp. 416–422, Oct. 2008.

[45] P. Comon and B. Mourrain, "Decomposition of quantics in sums of powers of linear forms," *Signal Process.*, vol.53, nos. 2–3, pp. 93–107, Sep. 1996.

[46] P. Comon, G. Golub, L. H. Lim, and B. Mourrain, "Symmetric tensors and symmetric tensor rank," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1254–1279, Sep. 2008.

[47] H. A. L. Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometr. Soc.*, vol. 14, no. 3, pp. 105–122, Jan. 2000.

[48] R. A. Horn, "The Hadamard product," in *Proc. Symp. Appl. Math*, vol. 40, Apr. 1990, pp. 1–7.

[49] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Tag recommendations based on tensor dimensionality reduction," in *Proc. ACM Conf. Recommender Syst.*, Lausanne, Switzerland, Oct. 2018, pp. 43–50.

[50] S. Liu and G. Trenkler, "Hadamard, Khatri-Rao, Kronecker and other matrix products," *Int. J. Inf. Syst. Sci.*, vol. 4, no. 1, pp. 160–177, 2008.

[51] K. J. Pearson and T. Zhang, "On spectral hypergraph theory of the adjacency tensor," *Graphs Comb.*, vol. 30, no. 5, pp. 1233–1248, Sep. 2014.

[52] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph Fourier transform," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 26–31.

[53] M. A. Henning and Y. Anders, "2-Colorings in k-regular k-uniform hypergraphs," *Eur. J. Comb.*, vol. 34, no. 7, pp. 1192–1202, Oct. 2013.

[54] E. Robeva, "Orthogonal decomposition of symmetric tensors," *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 1, pp. 86–102, Jan. 2016.

[55] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[56] D. Silva and A. Pereira, *Tensor Techniques for Signal Processing: Algorithms for Canonical Polyadic Decomposition*, Univ. Grenoble Alpes, Grenoble, France, 2016.

[57] V. Nguyen, K. Abed-Meraim, and N. Linh-Trung, "Fast tensor decompositions for big data processing," in *Proc. Int. Conf. Adv. Technol. Commun.*, Hanoi, Vietnam, Oct. 2016, pp. 215–221.

[58] T. Kolda, "Symmetric orthogonal tensor decomposition is trivial," *arXiv:1503.01375 [math. NA]*, Mar. 2015.

[59] S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd ed. New York, NY, USA: Academic, 2008.

[60] V. Rödl and J. Skokan, "Regularity lemma for k-uniform hypergraphs," *Random Struct. Algorithms*, vol. 25, no. 1, pp. 1–42, Jun. 2004.

[61] Z. K. Zhang and C. Liu, "A hypergraph model of social tagging networks," *J. Stat. Mech. Theory Exp.*, vol. 2010, no. 10, Oct. 2010, Art. no. P10005.

[62] K. J. Pearson and T. Zhang, "Eigenvalues on the adjacency tensor of products of hypergraphs," *Int. J. Contemp. Math. Sci*, vol. 8, no. 4, pp. 151–158, Jan. 2013.

[63] J. M. Mendel, "Use of higher-order statistics in signal processing and system theory: An update," *Adv. Algorithms Architect. Signal Process.*, vol. 975, pp. 126–145, Feb. 1988.

[64] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications," *Proc. IEEE*, vol. 79, no. 3, pp. 278–305, Mar. 1991.

[65] A. K. Nandi, *Blind Estimation Using Higher-Order Statistics*. New York, NY, USA: Springer, 2013.

[66] M. S. Choudhury, S. L. Shah, and N. F. Thornhill, "Diagnosis of poor control-loop performance using higher-order statistics," *Automatica*, vol. 40, no. 10, pp. 1719–1728, Oct. 2004.

[67] O. A. Dobre, Y. Bar-Ness, and W. Su, "Higher-order cyclic cumulants for high order modulation classification," in *Proc. IEEE Mil. Commun. Conf.*, Boston, MA, USA, Oct. 2003, pp. 112–117.

[68] M. B. Priestley, *Spectral Analysis and Time Series*. London, U.K.: Academic, 1981.

[69] A. Swami and J. M. Mendel, "Time and lag recursive computation of cumulants from a state-space model," *IEEE Trans. Autom. Control*, vol. 35, no. 1, pp. 4–17, Jan. 1990.

[70] M. Hein, S. Setzer, L. Jost, and S. S. Rangapuram, "The total variation on hypergraphs—Learning on hypergraphs revisited," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2013, pp. 2427–2435.

[71] G. Chen, J. Zhang, F. Wang, C. Zhang, and Y. Gao, "Efficient multi-label classification with hypergraph regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1658–1665.

[72] V. I. Paulsen and R. R. Smith, "Multilinear maps and tensor norms on operator systems," *J. Funct. Anal.*, vol. 73, no. 2, pp. 258–276, Aug. 1987.

[73] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Adaptive graph filtering: Multiresolution classification on graphs," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, Austin, TX, USA, Feb. 2014, pp. 427–430.

[74] S. Chen, F. Cerda, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovačević, "Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2879–2893, Mar. 2014.

[75] S. McCanne and M. J. Demmer, "Content-based segmentation scheme for data compression in storage and transmission including hierarchical segment representation," U.S. Patent, 6 667 700, Dec. 2003.

[76] K. Sayood, *Introduction to Data Compression*. Burlington, MA, USA: Morgan Kaufmann, 2017.

[77] D. Salomon, *Data Compression: The Complete Reference*. London, U.K.: Springer, 2004.

[78] M. Yan *et al.*, "Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints," *Inf. Sci.*, vol. 278, pp. 34–55, Sep. 2014.

[79] C. Adjih, S. Y. Cho, and P. Jacquet, "Near optimal broadcast with network coding in large sensor networks," *arXiv:0708.0975 [cs. NI]*, Aug. 2007.

[80] V. Zlatić, G. Ghoshal, and G. Caldarelli, "Hypergraph topological quantities for tagged social networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 3, Sep. 2009, Art. no. 036118.

[81] I. Konstas and M. Lapata, "Unsupervised concept-to-text generation with hypergraphs," in *Proc. Conf. North Amer. Assoc. Comput. Linguist. Human Lang. Technol.*, Montreal, QC, Canada, Jun. 2012, pp. 752–761.

[82] H. Boley, "Directed recursive labelnode hypergraphs: A new representation-language," *Artif. Intell.*, vol. 9, no. 1, pp. 49–85, Aug. 1977.

[83] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19, Jan. 2016, doi: 10.1145/2827872.

[84] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6163–6166.

[85] L. Qi, "Eigenvalues and invariants of tensors," *J. Math. Anal. Appl.*, vol. 325, no. 2, pp. 1363–1377, Jan. 2007.

[86] L.-H. Lim, "Singular values and eigenvalues of tensors: A variational approach," in *Proc. 1st IEEE Int. Workshop Comput. Adv. Multi Sensor Adapt. Process.*, Puerto Vallarta, Mexico, Dec. 2005, pp. 129–132.
[87] M. Ng, L. Qi, and G. Zhou, "Finding the largest eigenvalue of a nonnegative tensor," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 3, pp. 1090–1099, Feb. 2009.
[88] D. Cartwright and B. Sturmfels, "The number of eigenvalues of a tensor," *Linear Algebra Appl.*, vol. 438, no. 2, pp. 942–952, Jan. 2013.
[89] M. N. Marshall, "Sampling for qualitative research," *Family Practice*, vol. 13, no. 6, pp. 522–526, Dec. 1996.
[90] G. E. Batley and D. Gardner, "Sampling and storage of natural waters for trace metal analysis," *Water Res.*, vol. 11, no. 9, pp. 745–756, 1977.
[91] U. V. Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
[92] J. Jung, S. Chun, and K.-H. Lee, "Hypergraph-based overlay network model for the Internet of Things," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Milan, Italy, Dec. 2015, pp. 104–109.
[93] J. Yu, D. Tao, and M. Wang, "Adaptive hypergraph learning and its application in image classification," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3262–3272, Jul. 2012.
[94] S. Rital, H. Cherifi, and S. Miguet, "Weighted adaptive neighborhood hypergraph partitioning for image segmentation," in *Proc. Int. Conf. Pattern Recognit. Image Anal.*, Berlin, Germany, Aug. 2005, pp. 522–531.
[95] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 1–4.

Songyang Zhang received the Bachelor of Science degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA, USA.

His current research interests include hypergraph signal processing, behavior analysis over high-dimensional graphs, and learning over graphs.

Zhi Ding (S'88–M'90–SM'95–F'03) received the Ph.D. degree in electrical engineering from Cornell University, Ithaca, NY, USA, in 1990.

He is a Professor of electrical and computer engineering with the University of California at Davis, Davis, CA, USA. From 1990 to 2000, he was a Faculty Member with Auburn University, Auburn, AL, USA, and later, University of Iowa, Iowa City, IA, USA. He has held visiting positions in Australian National University, Canberra, ACT, Australia; the Hong Kong University of Science and Technology, Hong Kong; NASA Lewis Research Center, Cleveland, OH, USA; and USAF Wright Laboratory, New Haven, CT, USA. He has active collaboration with researchers from universities in Australia, Canada, China, Finland, Hong Kong, Japan, South Korea, Singapore, and Taiwan. He has coauthored the book titled: *Modern Digital and Analog Communication Systems*, (Oxford University Press, 2019, 5th edition).

Prof. Ding was a recipient of the IEEE Communication Society's WTC Award in 2012. He has been serving on technical programs of several workshops and conferences. He was an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1994 to 1997 and from 2001 to 2004 and IEEE SIGNAL PROCESSING LETTERS from 2002 to 2005. He was a member of technical committee on Statistical Signal and Array Processing and technical committee on Signal Processing for Communications from 1994 to 2003. He was the General Chair of the 2016 IEEE International Conference on Acoustics, Speech, and Signal Processing and the Technical Program Chair of the 2006 IEEE Globecom. He was also an IEEE Distinguished Lecturer (Circuits and Systems Society, from 2004 to 2006; Communications Society, from 2008 to 2009). He served on the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS Steering Committee Member from 2007 to 2009 and its Chair from 2009 to 2010.

Shuguang Cui (S'99–M'05–SM'12–F'14) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2005.

He has been an Assistant, an Associate, a Full, and the Chair Professor with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA; Texas A&M University, Uvalde, TX, USA; University of California at Davis, Davis, CA, USA; and the Chinese University of Hong Kong, Shenzhen. He has also been the Vice Director with the Shenzhen Research Institute of Big Data, Shenzhen, China. His current research interests include data driven large-scale system control and resource management, large data set analysis, Internet of Things system design, energy harvesting-based communication system design, and cognitive network optimization.

Prof. Cui was a recipient of the IEEE Signal Processing Society 2012 Best Paper Award, the Amazon AWS Machine Learning Award in 2018, and the Highly Cited Researcher Award by Thomson Reuters. He was listed in the Worlds' Most Influential Scientific Minds by ScienceWatch in 2014. He has served as the general co-chair and the TPC co-chair for many IEEE conferences. He has also been serving as the Area Editor for the *IEEE Signal Processing Magazine*, and an Associate Editor for the IEEE TRANSACTIONS ON BIG DATA, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Series on Green Communications and Networking, and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has been the Elected Member for IEEE Signal Processing Society SPCOM Technical Committee from 2009 to 2014 and the Elected Chair for IEEE ComSoc Wireless Technical Committee from 2017 to 2018. He is a member of the Steering Committee for the IEEE TRANSACTIONS ON BIG DATA and the Chair of the Steering Committee for IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He was also a member of the IEEE ComSoc Emerging Technology Committee. He was elected as an IEEE ComSoc Distinguished Lecturer in 2014 and IEEE VT Society Distinguished Lecturer in 2019.