

t-HGSP: Hypergraph Signal Processing Using t-Product Tensor Decompositions

Karelia Pena-Pena , Member, IEEE, Daniel L. Lau , Senior Member, IEEE,
and Gonzalo R. Arce , Life Fellow, IEEE

Abstract—Graph signal processing (GSP) techniques are powerful tools that model complex relationships within large datasets, being now used in a myriad of applications in different areas including data science, communication networks, epidemiology, and sociology. Simple graphs can only model pairwise relationships among data which prevents their application in modeling networks with higher-order relationships. For this reason, some efforts have been made to generalize well-known graph signal processing techniques to more complex graphs such as hypergraphs, which allow capturing higher-order relationships among data. In this article, we provide a new hypergraph signal processing framework (t-HGSP) based on a novel tensor-tensor product algebra that has emerged as a powerful tool for preserving the intrinsic structures of tensors. The proposed framework allows the generalization of traditional GSP techniques while keeping the dimensionality characteristic of the complex systems represented by hypergraphs. To this end, the core elements of the t-HGSP framework are introduced, including the shifting operators and the hypergraph signal. The hypergraph Fourier space is also defined, followed by the concept of bandlimited signals and sampling. In our experiments, we demonstrate the benefits of our approach in applications such as clustering and denoising.

Index Terms—Graph, Signal Processing, Tensor, Data Analysis.

I. INTRODUCTION

GRAPH signal processing (GSP) has recently emerged as a rich theoretical framework to analyze and process signals that live on irregular grids — a trait that commonly arises in a broad range of applications including biology [1], neuroscience [2], urban transportation [3], sensors, and social networks [4]. Nodes of a graph may represent individuals in social networks, brain regions in neural networks, or transactions in financial networks. Classical Digital Signal Processing (DSP) principles such as sampling, filtering, compression, denoising,

Manuscript received 16 August 2022; revised 31 January 2023 and 2 April 2023; accepted 1 May 2023. Date of publication 16 May 2023; date of current version 25 May 2023. This work was supported in part by the National Science Foundation under Grants CCF 2230161 and 2230162, in part by APOSER under Grant FA9550-22-1-0362, and in part by the Institute of Financial Services Analytics at the University of Delaware. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dorina Thanou. (Corresponding author: Gonzalo Arce.)

Karelia Pena-Pena and Gonzalo R. Arce are with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: kareliap@udel.edu; arce@udel.edu).

Daniel L. Lau is with the Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506 USA (e-mail: dllau@uky.edu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TSIPN.2023.3276687>, provided by the authors.

Digital Object Identifier 10.1109/TSIPN.2023.3276687

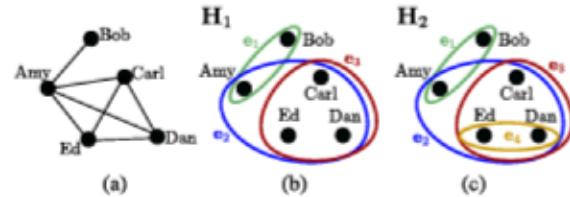


Fig. 1. In a co-authorship network, two different hypergraphs (b) H_1 (c) H_2 are mapped to the same simple graph in (a) by the clique expansion. Hyperedges are color-coded by publication, e.g. the red hyperedge (e_3) indicates that Carl, Dan, and Ed coauthored a publication. Since $e_4 \subset e_3$, e_4 is ignored in the clique expansion representation.

and signal representations have thus been extended for signals on graphs under the umbrella of GSP [5], [6], [7].

At the heels of GSP, a new and exciting set of tools for processing data on complex structures is emerging, coined *Hypergraph Signal Processing* (HGSP) [8], [9], [10], [11], [12]. The motivation lies *first* in the ability of hypergraphs to capture high-order interactions between more than two nodes. In the case of a co-authorship network, for instance, a hypergraph can better represent the relationships between the different groups of authors. While edges in conventional graphs (Fig. 1(a)) can only model pairwise relationships between nodes, limiting GSP to single-way analysis, hyperedges connect more than two nodes as illustrated in Fig. 1(b)–(c). Hypergraph signals are thus defined as those associated to the vertices of a hypergraph, whose polyadic interactions are modeled by hyperedges. The literature on data processing in high-order networks is sparse compared to GSP [12], even though higher-order relationships are widespread in many complex systems such as multiple neurons firing at the same time [13], biochemical reactions having more than two proteins [1], people interacting in small groups of more than two people [14], and multilateral relationships among several related points (e.g., on a surface) in 3D point cloud [15].

A *second* motivation for advancing hypergraph signal processing is to overcome the limitation of standard graphs to model only single-layer connections. Many complex networks in nature exhibit multiple-layer interactions between two nodes, which can be modeled by multilayer graphs with each type of interaction defining a layer [16], [17], [18]. Mathematically, the matrix algebra of graphs cannot effectively model multi-layer relationships as all entries in a matrix are considered equivalent [9]; however, networks with multi-layer relationships have been successfully studied by being mapped to hypergraphs [19]. Among the applications of multilayer graphs are multi-sensor

point clouds, imaging spectroscopy, and multilayer social networks. Point cloud processing, in particular, is an area of significant growth as it arises with new sensor technologies such as LiDAR [20], tomographic synthetic aperture radar [21], and time-of-flight spectral cameras [22], [23], [24]. Point clouds are ubiquitous in computer vision and are essential in the exploration of planetary bodies [25], [26].

The advantages of HGSP, however, are currently hindered by several technical challenges including:

1. *Choosing an appropriate algebraic descriptor:* In GSP, computation is enabled by encoding the graph structure in an adjacency matrix or its associated Laplacian. Hypergraphs, however, have many different algebraic descriptors for the same higher-order interaction structure, and choosing an appropriate algebraic descriptor is a key element when defining an HGSP framework. In fact, hypergraphs can be described by both matrices [27] and tensors [28]. GSP operations can easily be applied to hypergraph matrix-based descriptors. Nevertheless, these fail to capture the high-order structure of the data. Higher dimensional tensors, on the other hand, allow capturing deeper insights from high-order data.

2. *Defining a hypergraph signal shifting operation:* In GSP, the algebraic descriptors of the graph are called *graph shift operators* which are a generalization of the classical time delay in DSP. The shifting operation replaces a signal value at each node of a graph with a linear combination of the signal values at the neighbors of that node. When defining a new tensor-based HGSP framework, modeling the hypergraph signal interactions over hyperedges that connect more than two nodes is not as straightforward as in the graph case.

3. *Defining a loss-free hypergraph Fourier transform:* In GSP, the graph Fourier transform is defined based on the eigendecomposition of the *graph shift operator*. In HGSP when considering tensor algebraic descriptors, the hypergraph Fourier transform is not uniquely generalized since the concept of eigendecomposition of a matrix is not uniquely defined for high-order tensors. Previous efforts relied on an approximated, yet computationally intensive decomposition, that leads to subspace mappings of the signal, hence, providing a lossy Fourier transform.

This article tackles these limitations by proposing a new HGSP framework based on t-product factorizations. The main contributions of this article can be enumerated as follows: (i) formulate the core elements of the new HGSP framework, which includes the tensor-based algebraic representation of hypergraphs, hypergraph signals, and the hypergraph shifting operation; (ii) define a loss-free hypergraph Fourier transform using t-product factorizations; (iii) provide an analysis of frequency and the concept of bandlimited signals by defining a measure of the total variation of signals on hypergraphs; (iv) introduce the fundamentals for filtering, sampling, and recovery of hypergraph signals.

II. BACKGROUND AND RELATED WORK

A. Graph Signal Processing Overview

An undirected simple graph on N data points is given by $G = (V(G), E(G))$, where $V(G)$ is the set of nodes defined on the N

data points and $E(G)$ is the set of edges describing the pairwise interactions between these nodes. These interactions can also be represented by the symmetric adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ whose entries are $\mathbf{A}(u, v) = 1$ if there is an edge connecting the nodes u and v and 0 otherwise. The combinatorial graph Laplacian associated with G and computed as $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is a symmetric positive semidefinite matrix where the diagonal matrix \mathbf{D} stores the degrees of the nodes in the graph and its entries are given by $\mathbf{D}(u, u) = \sum_{v \in V(G)} \mathbf{A}(u, v)$. A real valued signal, $\mathbf{x} \in \mathbb{R}^N$, on the graph, G , is defined as a function $\mathbf{x} : V(G) \rightarrow \mathbb{R}$. The component $\mathbf{x}(v)$ represents the value of the signal on the node $v \in V(G)$. From $G = (V(G), E(G))$, GSP defines a *shift operator*, \mathbf{F} , as a local operation that replaces the signal's value at each node with a linear combination of the signal's values from neighboring nodes according to $\mathbf{F}\mathbf{x}$. Several *graph shift operators* such as the adjacency or the Laplacian can be considered in order to exploit different properties of the graph and to define a *graph signal processing framework* [6], [29].

The notion of linear filtering the graph signal, \mathbf{x} , by the filter, \mathbf{Q} , is achieved by multiplication $\mathbf{y} = \mathbf{Q}\mathbf{x}$, resulting in a new graph signal \mathbf{y} . While \mathbf{Q} can be any arbitrary matrix, it is considered a *linear shift invariant* (LSI) operator if it satisfies the condition that $\mathbf{Q}\mathbf{F}\mathbf{x} = \mathbf{F}\mathbf{Q}\mathbf{x}$. Now since \mathbf{F} is real and symmetric, it is diagonalizable by means of the eigenvalue decomposition as $\mathbf{F} = \mathbf{V}\Lambda\mathbf{V}^\top$ where \mathbf{V} is a unitary matrix whose column vectors are eigenvectors and Λ is a diagonal matrix of corresponding eigenvalues [30], [31]. Then, given a filtering weight function $q : \mathbb{R} \rightarrow \mathbb{R}$, a *linear shift invariant* filter can be written as $\mathbf{Q} = \sum_{k=1}^N q(\lambda_k) \mathbf{v}_k \mathbf{v}_k^\top = \mathbf{V}q(\Lambda)\mathbf{V}^\top$, where $q(\Lambda) = \text{diag}(q(\lambda_1), \dots, q(\lambda_N))$ is the frequency response of the filter \mathbf{Q} . By spanning the set \mathbb{R}^N such that any graph signal, \mathbf{x} , can be written as a linear combination of column vectors from \mathbf{V} , the *Graph Fourier Transform* (GFT) of a signal \mathbf{x} on G can then be defined as $\hat{\mathbf{x}} = \mathbf{V}^\top \mathbf{x}$ and its inverse GFT as $\mathbf{x} = \mathbf{V}\hat{\mathbf{x}}$ with the eigenvalues interpreted as frequencies [32]. As N -length vectors, the columns of \mathbf{V} are, themselves, graph signals whose corresponding eigenvalues are measures of their total variations. The bandwidth of \mathbf{x} is defined in terms of the nonzero components of $\hat{\mathbf{x}}$.

B. From Graphs to Hypergraphs

A hypergraph $\mathbf{H} = (V(\mathbf{H}), E(\mathbf{H}))$ is defined as the pair of one set of nodes $V(\mathbf{H}) = \{\mathbf{v}_1, \dots, \mathbf{v}_N\}$ and a set of edges $E(\mathbf{H}) = \{\mathbf{e}_1, \dots, \mathbf{e}_K\}$ whose elements, different from simple graphs, are multi-element subsets of $V(\mathbf{H})$ called hyperedges. Let $M = \max\{|\mathbf{e}_i| : \mathbf{e}_i \in E(\mathbf{H})\}$ be the maximum cardinality of the hyperedges, shorted as $m.c.e(\mathbf{H})$. $\mathbf{H} = (V(\mathbf{H}), E(\mathbf{H}))$ can be represented using matricial or tensorial algebraic descriptors. In the matrix-based approach, a hypergraph is represented by the incidence matrix $\mathbf{B} \in \mathbb{R}^{N \times E}$ such that $\mathbf{B}_{ue} = 1$ if the vertex u is an element in the hyperedge e and 0 otherwise. Even though the matrix-based representation of the hypergraph is simple, it fails to capture the high-dimensionality structure of the hypergraph. For this reason, in this work, we use the tensorial representation of hypergraphs to propose a new HGSP framework.

The adjacency tensor of a hypergraph $\mathbf{H} = (V(\mathbf{H}), E(\mathbf{H}))$ with N nodes and $M = m.c.e(\mathbf{H})$ is traditionally represented by an M th-order N -dimensional tensor $\mathcal{A} \in \mathbb{R}^{N^M}$ defined as $\mathcal{A} = a_{p_1, p_2, \dots, p_M}$, $1 \leq p_1, p_2, \dots, p_M \leq N$. A hyperedge $e_t = \{\mathbf{v}_{l_1}, \mathbf{v}_{l_2}, \dots, \mathbf{v}_{l_{c_t}}\} \in E(\mathbf{H})$ of cardinality $c_t = |e_t| \leq M$, is represented in \mathcal{A} by $a_{p_1, p_2, \dots, p_M} = c_t/\alpha_t$, where

$$\alpha_t = \sum_{k_1, k_2, \dots, k_{c_t} \geq 1, \sum_{j=1}^{c_t} k_j = M} \frac{M!}{k_1! k_2! \dots k_{c_t}!}, \quad (1)$$

and c_t out of the M indices in the set $\{p_1, p_2 \dots p_M\}$ are exactly the same as $\{l_1, l_2, \dots, l_{c_t}\}$ and for the other $M - c_t$ indices every possible subset combination from $\{l_1, l_2, \dots, l_{c_t}\}$ is taken. Observe that \mathbf{l} enumerates all the possible combinations of c_t positive integers $\{k_1, k_2, \dots, k_{c_t}\}$ whose summation satisfies $\sum_{j=1}^{c_t} k_j = M$. Note that a hypergraph with $M = 2$ degrades to a simple graph with a binary adjacency matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$; hence, HGSP is a generalization of GSP.

The degree of a vertex $\mathbf{v}_k \in V(\mathbf{H})$ is the number of hyperedges containing \mathbf{v}_k , i.e., $d(\mathbf{v}_k) = \sum_{t_2, t_3, \dots, t_m=1}^N a_{k, t_2, t_3, \dots, t_m}$. Then, the traditional **Laplacian tensor** is defined as $\mathcal{L} = \mathcal{D} - \mathcal{A} \in \mathbb{R}^{N^M}$, where, $\mathcal{D} \in \mathbb{R}^{N^M}$ is the **degree tensor** defined as $\mathcal{D} = d_{p_1, p_2, \dots, p_M}$, $1 \leq p_1, p_2, \dots, p_M \leq N$, which is a super-diagonal tensor with diagonal elements $d_{\underbrace{k, \dots, k}_{M \text{ times}}} = d(\mathbf{v}_k)$, $1 \leq k \leq N$, and zero otherwise.

C. Related Work

The theory of signal processing on higher-order networks is largely unexplored compared to that on graphs. The most prominent abstractions for such polyadic data are simplicial complexes [33] and hypergraphs [11], [34]. Simplicial complexes are used as an alternative representation of hypergraphs, with the particular drawback of the inclusion property, characteristic of simplicial complexes, that is undesirable when representing interactions that are exclusive to multiple nodes but do not imply the interaction between all the subsets of nodes [11], [12]. For instance, in a co-authorship network (Fig. 1), having a paper of three or more authors does not imply that these authors have written papers in pairs. While tensor-based hypergraph representations can differentiate these two cases, simple graphs, simplicial complexes, and even matrix-based hypergraph representations, which represent hypergraphs as graphs, cannot [12]. As shown in Fig. 1, a common matrix-based hypergraph representations, the clique expansion, replaces every hyperedge with a clique subgraph, clearly failing to capture lower-dimensional relationships and not providing an injective mapping for a hypergraph.

Recently, a tensor-based hypergraph signal processing framework was introduced [9], which proceeds analogously to GSP. In [9], much like the Fourier Transform in GSP is defined via the eigen-decomposition of the graph Laplacian or adjacency matrix, the Fourier transform in HGSP is defined via the canonical polyadic (CP) tensor decomposition of the hypergraph adjacency or Laplacian tensor. The canonical polyadic (CP) decomposition expresses a tensor as the finite sum of rank-one tensors (Fig. 2(a)). For instance, the CP decomposition of a third-order tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is represented

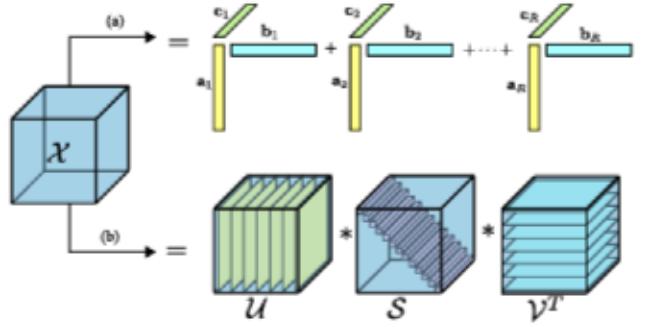


Fig. 2. Tensor \mathcal{X} and its (a) CP decomposition, and (b) t-SVD decomposition.

as $\mathcal{X} = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ where $\mathbf{a}_r \in \mathbb{R}^{N_1}$, $\mathbf{b}_r \in \mathbb{R}^{N_2}$, and $\mathbf{c}_r \in \mathbb{R}^{N_3}$, and \circ represents the tensor outer product operation. Particularly, the symmetric orthogonal-CP decomposition [35], being a special case of the CP decomposition, is used in [9] to decompose a super-symmetric tensor $\mathcal{F} \in \mathbb{R}^{N \times N \times N}$ representing either the hypergraph adjacency or Laplacian tensor as $\mathcal{F} \approx \sum_{r=1}^R \lambda_r \mathbf{f}_r \circ \mathbf{f}_r \circ \mathbf{f}_r$, where the \mathbf{f}_r 's form an orthogonal set. The integer R is the smallest number of rank-one tensors required to express \mathcal{F} and is referred to as the CP rank of \mathcal{F} [36]. The CP rank has no relaxation which limits its applications. While the CP-tensor decomposition approach to HGSP was shown effective in some image processing applications [9], [10], it has some drawbacks. The first drawback comes from the fact that the hypergraph spectrum space is obtained from the orthogonal-CP decomposition of the hypergraph adjacency tensor \mathcal{A} , and, as shown in Fig. 3(a), this tensor does not have an exact CP orthogonal decomposition. In fact, most tensors generally do not have an exact CP orthogonal decomposition [37]. Current efforts have focused on finding symmetric orthogonal approximations [35]; however, errors add uncertainty, affecting the accuracy of the hypergraph Fourier transform which is undesirable, particularly when filtering on the Fourier domain. Second, finding the CP-rank R is an NP-hard problem which invariably poses high computational complexity [38]. Third, generally, the rank R is smaller than the number of nodes in the hypergraph N [9]. As a consequence, hypergraph signals in \mathbb{R}^N are invariably mapped to a subspace of \mathbb{R}^N which does not guarantee perfect recovery on the inverse hypergraph Fourier transform for any type of signal as depicted in Fig. 3(a).

An alternative setting for hypergraph signal processing that can overcome all of the above-mentioned limitations is therefore needed. Choosing the tensor decomposition that better parallels the characteristics of the eigendecomposition in traditional GSP frameworks becomes a key aspect when defining an HGSP framework. To this end, we adopt an alternative setting based on the more recently proposed tensor-tensor multiplication (t-product), in which the familiar tools of linear algebra are extended to better understand tensors [39]. The t-product of two small tensors $\mathbf{a} \in \mathbb{R}^{1 \times 1 \times N_3}$ and $\mathbf{b} \in \mathbb{R}^{1 \times 1 \times N_3}$, known as tubal scalars, is the tensor $\mathbf{c} \in \mathbb{R}^{1 \times 1 \times N_3}$ computed as $\mathbf{c} = \mathbf{a} * \mathbf{b}$, where $*$ denotes the circular convolution of two vectors. Thus, by considering tensors as matrices of tubal scalars, tensor factorizations built from t-products are analog to matrix factorizations

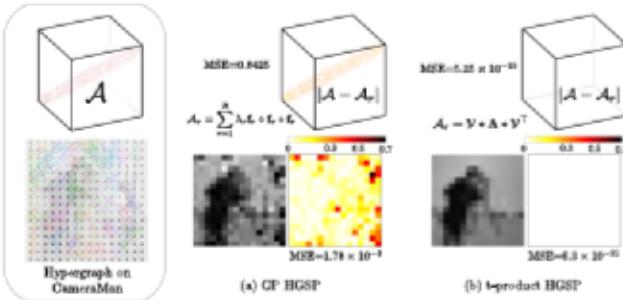


Fig. 3. (Bottom-left) Hypo-graph on the 16×16 CameraMan image built using the method in [41] and its (top-left) 3rd-order adjacency tensor representation \mathcal{A} . At the bottom of (a) and (b) is the signal after applying the forward and inverse hypergraph Fourier transform and the corresponding absolute error using: the symmetric orthogonal CP-based decomposition, and the proposed t-product factorization, respectively. At the top of (a) and (b) are the absolute errors of the tensor decompositions. The advantages of the proposed approach are readily seen.

such as the SVD, QR, and eigendecompositions [39]. The tensor singular value decomposition, based on this t-product, was coined t-SVD. The significance of these decompositions is that they allow for the extension of familiar matrix analysis to the multilinear setting while preserving the intrinsic structure of tensors and avoiding the loss of information. The algebraic framework for higher-order tensors is thus analogous to that of matrices such as the transpose, identity, orthogonality, and diagonalization. The t-SVD of a third-order tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is for instance $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ where $*$ denotes the t-product [39], $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ and $\mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$ are orthogonal tensors, and $\mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is an f-diagonal tensor where the frontal slices are diagonal matrices as shown in Fig. 2(b). Although the t-SVD was first demonstrated on 3rd-order tensors, it is easily extended to n -th ($n > 3$) order tensors [40]. The proposed research exploits this novel set of t-product factorizations to develop a new HGSP framework, dubbed as t-HGSP, that aims at being more stable and, more importantly, loss-free compared to previous tensor-based HGSP frameworks as demonstrated in Fig. 3(b). To formulate the t-product and its algebra as the backbone of our t-HGSP framework the following definitions need to be introduced.

D. t-Product Operations

Denoting vectors by bold lowercase letters (e.g. \mathbf{a}), matrices as uppercase letters (e.g. \mathbf{A}), and tensors as calligraphic letters (e.g. \mathcal{A}), we first define the (i, j) -th tube scalar of the 3rd-order tensor \mathcal{A} as \mathbf{a}_{ij} which is illustrated in Fig. 4(b). The j -th lateral slice of the tensor, shown in Fig. 4(c), is denoted as $\tilde{\mathcal{A}}_j \equiv \mathcal{A}(:, j, :)$ $\in \mathbb{R}^{N_1 \times 1 \times N_3}$ which is a vector of tubal scalars. The k -th frontal slice depicted in Fig. 4(d) is denoted as $\mathbf{A}^{(k)} \equiv \mathcal{A}(:, :, k)$ $\in \mathbb{R}^{N_1 \times N_2 \times 1}$ which is a matrix.

Definition 1 (t-product [40], [42]):

As mentioned before, the t-product of two tubal scalars $\mathbf{a} \in \mathbb{R}^{1 \times 1 \times N_3}$ and $\mathbf{b} \in \mathbb{R}^{1 \times 1 \times N_3}$ is also a tubal scalar $\mathbf{c} \in \mathbb{R}^{1 \times 1 \times N_3}$ computed as $\mathbf{c} = \mathbf{a} * \mathbf{b}$, where $*$ denotes the circular convolution of two vectors. Then, the t-product of two vectors of tubal

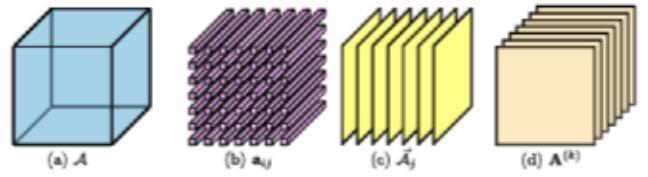


Fig. 4. (a) Third-order tensor \mathcal{A} divided into: (b) a matrix of tubal scalars, (c) lateral slices (vector of tubal scalars), and (d) frontal slices.

scalars $\tilde{\mathcal{A}} \in \mathbb{R}^{N_1 \times 1 \times N_3}$ and $(\tilde{\mathcal{B}})^T \in \mathbb{R}^{1 \times N_2 \times N_3}$ is a matrix of tubal scalars $\mathcal{C} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ computed as

$$\mathcal{C} = \tilde{\mathcal{A}} * (\tilde{\mathcal{B}})^T = \begin{bmatrix} \mathbf{a}_{1,1} * \mathbf{b}_{1,1} & \cdots & \mathbf{a}_{1,1} * \mathbf{b}_{1,N_2} \\ \vdots & \ddots & \vdots \\ \mathbf{a}_{N_1,1} * \mathbf{b}_{1,1} & \cdots & \mathbf{a}_{N_1,1} * \mathbf{b}_{1,N_2} \end{bmatrix}. \quad (2)$$

In general, the t-product of two 3rd-order tensors $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$ is the tensor $\mathcal{C} \in \mathbb{R}^{N_1 \times N_4 \times N_3}$ computed as

$$\begin{aligned} \mathcal{C} &= \mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})) \\ &= \text{fold} \left(\begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(N_3)} & \cdots & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \cdots & \mathbf{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(N_3)} & \mathbf{A}^{(N_3-1)} & \cdots & \mathbf{A}^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{B}^{(1)} \\ \mathbf{B}^{(2)} \\ \vdots \\ \mathbf{B}^{(N_3)} \end{bmatrix} \right) \end{aligned} \quad (3)$$

where the operator $\text{bcirc}(\mathcal{A})$ converts the set of frontal slices of the tensor \mathcal{A} into a block circulant matrix and $\text{unfold}(\mathcal{B})$ stacks vertically the set of frontal slices of \mathcal{B} into a $N_2 N_3 \times N_4$ matrix. The operator $\text{fold}()$ reverses this process, $\text{fold}(\text{unfold}(\mathcal{A})) = \mathcal{A}$. Circulant matrices are diagonalized by the discrete Fourier transform; hence, the t-product can be computed efficiently in the Fourier domain as explained in [42]. Using MATLAB notation, let $\hat{\mathcal{A}} := \text{fft}(\mathcal{A}, [], 3)$ denote the tensor obtained by applying the fast Fourier transform (FFT) along each tubal element of \mathcal{A} . For the remainder of this article, the hat notation refers to a tensor that has gone through this operation. Thus, the t-product of $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$ can also be computed by matrix multiplication of each pair of frontal slices in the Fourier Domain as

$$\mathcal{C} = \mathcal{A} * \mathcal{B} := \text{ifft} \left(\left[\hat{\mathcal{A}}^{(k)} \hat{\mathcal{B}}^{(k)} \right]_{k=1}^{N_3}, [], 3 \right). \quad (4)$$

The t-product can be easily extended to high-order tensors in a recursive manner. The t-product of $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3 \times \cdots \times N_p}$ and $\mathcal{B} \in \mathbb{R}^{N_2 \times L \times N_3 \times \cdots \times N_p}$ is an order- p ($p > 3$) tensor $\mathcal{C} \in \mathbb{R}^{N_1 \times L \times N_3 \times \cdots \times N_p}$ computed recursively as

$$\mathcal{C} = \mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) * \text{unfold}(\mathcal{B}))$$

$$= \text{fold} \left(\begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(N_p)} & \dots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \dots & \mathcal{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{(N_p)} & \mathcal{A}^{(N_p-1)} & \dots & \mathcal{A}^{(1)} \end{bmatrix} * \begin{bmatrix} \mathcal{B}^{(1)} \\ \mathcal{B}^{(2)} \\ \vdots \\ \mathcal{B}^{(N_p)} \end{bmatrix} \right) \quad (5)$$

where $\mathcal{A}^{(l)}$ and $\mathcal{B}^{(l)}$ are order- $(p-1)$ tensors formed from holding the p -th index at l , respectively for $l = 1, 2, \dots, N_p$ [40]. Thus, each successive t-product operation involves tensors of one order less and at the base label of recursion there is a t-product of 3rd-order tensors.

Given that the extension to high-order tensors is obtained by recursion, for simplicity and without loss of generality, we present our framework for the base case which is given by hypergraphs with $M = 3$ whose adjacency tensor is a 3rd-order tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$.

Definition 2 (Transpose and Symmetric tensors [42]): The transpose of a 3rd-order tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, denoted as \mathcal{A}^\top , is the tensor obtained by transposing each of the frontal slices and then reversing the order of the transposed frontal slices 2 through N_3 . For a higher-order tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times \dots \times N_p}$, its transpose $\mathcal{A}^\top \in \mathbb{R}^{N_2 \times N_1 \times \dots \times N_p}$ is obtained by recursively transposing each $\mathcal{A}^{(l)}$ for $l = 1, 2, \dots, N_p$ and then reversing the order of the $\mathcal{A}^{(l)}$'s from $l = 2$ to $l = N_p$ as

$$\mathcal{A}^\top = \text{fold} \left(\begin{bmatrix} \mathcal{A}^{(1)\top} \\ \mathcal{A}^{(N_p)\top} \\ \vdots \\ \mathcal{A}^{(2)\top} \end{bmatrix} \right). \quad (6)$$

The tensor \mathcal{A} is symmetric if $\mathcal{A} = \mathcal{A}^\top$.

Definition 3 (Ordering [43]): Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{1 \times 1 \times N_3}$ be two tubal scalars then

$$\begin{aligned} \mathbf{a} \leq \mathbf{b} & \text{ when } \hat{a}^{(k)} \leq \hat{b}^{(k)} \quad \forall k, \\ \mathbf{a} < \mathbf{b} & \text{ when } \hat{a}^{(k)} < \hat{b}^{(k)} \quad \forall k, \end{aligned} \quad (7)$$

that is $\mathbf{a} \leq \mathbf{b}$ if and only if each of the components of \mathbf{a} in the Fourier domain $\hat{a}^{(k)}$ are pair-wise less than or equal than the components of \mathbf{b} in the Fourier domain $\hat{b}^{(k)}$.

Definition 4 (Norm [43]): The norm of a vector of tubal scalars $\vec{\mathcal{X}} \in \mathbb{R}^{N_1 \times 1 \times N_3}$ is a $1 \times 1 \times N_3$ tubal scalar and it can be computed as

$$\|\vec{\mathcal{X}}\| = \left(\vec{\mathcal{X}}^\top * \vec{\mathcal{X}} \right)^{1/2} = \text{ifft} \left(\sqrt{\hat{\mathbf{X}}^{(k)\top} \hat{\mathbf{X}}^{(k)}} \Big|_{k=1}^{N_3}, \mathbf{0}, 3 \right), \quad (8)$$

where $\hat{\mathbf{X}}^{(k)} \in \mathbb{R}^{N_1 \times 1}$ is the k -th frontal slice of $\hat{\mathcal{X}} = \text{fft}(\vec{\mathcal{X}}, \mathbf{0}, 3)$.

Definition 5 (Inverse of a Tensor [42]): A 3rd-order $N_1 \times N_1 \times N_3$ tensor \mathcal{A} has \mathcal{B} as its inverse if

$$\mathcal{A} * \mathcal{B} = \mathcal{I}_{N_1 N_1 N_3}, \quad \text{and} \quad \mathcal{B} * \mathcal{A} = \mathcal{I}_{N_1 N_1 N_3} \quad (9)$$

where $\mathcal{I}_{N_1 N_1 N_3}$ is the identity tensor whose first frontal slice is the $N_1 \times N_1$ identity matrix, and the other frontal slices are all zeros.

Definition 6 (t-eigendecomposition [42]): Let $\mathcal{A} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$. If each frontal face of $\hat{\mathcal{A}}$ is diagonalizable, that is, $\hat{\mathbf{A}}^{(k)} = \hat{\mathbf{V}}^{(k)} \hat{\Lambda}^{(k)} (\hat{\mathbf{V}}^{(k)})^{-1}$ then \mathcal{A} has the following t-eigendecomposition form

$$\mathcal{A} = \mathcal{V} * \hat{\Lambda} * \mathcal{V}^{-1} \quad (10)$$

where $\hat{\Lambda} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ is an f-diagonal tensor, and the following holds

$$\mathcal{A} * \mathcal{V} = \mathcal{V} * \hat{\Lambda} \quad (11)$$

$$\mathcal{A} * \vec{\mathcal{V}}_j = \vec{\mathcal{V}}_j * \lambda_j \quad (12)$$

where $\lambda_j \in \mathbb{R}^{1 \times 1 \times N_3}$ and $\vec{\mathcal{V}}_j \in \mathbb{R}^{N_1 \times 1 \times N_3}$ corresponds to an eigen-tuple and its corresponding eigen-matrix. If \mathcal{A} is symmetric i.e., $\mathcal{A} = \mathcal{A}^\top$, then $\mathcal{A} = \mathcal{V} * \hat{\Lambda} * \vec{\mathcal{V}}$, where \mathcal{V} is an orthogonal tensor and the set of eigen matrices $\{\vec{\mathcal{V}}_1, \vec{\mathcal{V}}_2, \dots, \vec{\mathcal{V}}_{N_1}\}$ form an orthonormal set (Definition [13] in the Supplemental Material). This decomposition can be efficiently computed in the Fourier domain by means of matrix eigendecomposition as shown in Algorithm 1 (for 3rd-order tensors).

Definition 7 (Multi-Rank [42]): The multi-rank of a 3rd-order $N_1 \times N_2 \times N_3$ tensor \mathcal{A} is the tubal scalar $\rho(\mathcal{A}) \in \mathbb{R}^{1 \times 1 \times N_3}$ with frontal slices $\rho(\mathcal{A})^{(k)}$ being the rank of the k -th matrix $\hat{\mathbf{A}}^{(k)}$. Since $\hat{\mathbf{A}}^{(k)}$ is a $N_1 \times N_2$ matrix, $\rho(\mathcal{A})^{(k)} \leq \min(N_1, N_2)$. The multi-nullity is the complimentary tubal scalar $\eta(\mathcal{A}) \in \mathbb{R}^{1 \times 1 \times N_3}$ with entries $\eta(\mathcal{A})^{(k)} = N_2 - \rho(\mathcal{A})^{(k)}$.

III. HYPERGRAPH SHIFTING OPERATION

In order to make these t-product operations the backbone of our t-HGSP framework, we define the hypergraph shifting operation as

$$\vec{\mathcal{Y}} = \mathcal{F} * \vec{\mathcal{X}}, \quad (13)$$

where $\vec{\mathcal{X}}$ is the hypergraph signal, $\vec{\mathcal{Y}}$ is the one-time shifted/filtered signal and \mathcal{F} is the shifting operator. As in GSP, \mathcal{F} is any operator that captures the relational dependencies between nodes, including the adjacency tensor \mathcal{A} and the Laplacian \mathcal{L} . Motivated by the connection between the t-product and the Fourier transform [39] and the fact that, in this article, we are only considering undirected hypergraphs, we also define symmetric adjacency and Laplacian tensor descriptors. Note that, for $M > 2$, the tensors \mathcal{A} and \mathcal{L} introduced before are not symmetric according to Definition 2. Therefore, we define a symmetrization operator $\text{sym}(\mathcal{A})$ that generates a symmetric version $\mathcal{A}_s \in \mathbb{R}^{N \times N \times (2N+1)}$ of $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$, by adding a matrix of zeros $\mathbf{0}_{N \times N}$ as the first frontal slice, dividing by 2, and reflecting the frontal slices of \mathcal{A} along the third dimension

as

$$\mathcal{A}_s = \text{sym}(\mathcal{A}) = \text{fold} \left(\begin{bmatrix} 0_{N \times N} \\ \frac{1}{2}\mathcal{A}^{(1)} \\ \frac{1}{2}\mathcal{A}^{(2)} \\ \vdots \\ \frac{1}{2}\mathcal{A}^{(2)} \\ \frac{1}{2}\mathcal{A}^{(1)} \end{bmatrix} \right). \quad (14)$$

Now, if we let $N_s = 2N + 1$, for a higher-order tensor $\mathcal{A} \in \mathbb{R}^{N^M}$, its symmetric version is a M th-order tensor $\mathcal{A}_s \in \mathbb{R}^{N \times N \times N_s^{M-2}}$ obtained by recursively appending a $(M-1)$ th-order tensor of zeros \mathcal{O} at the front, dividing by 2, and reflecting the $(M-1)$ th-order tensors $\mathcal{A}^{(l)}$ along the last dimension as

$$\mathcal{A}_s = \text{sym}(\mathcal{A}) = \text{fold} \left(\begin{bmatrix} \mathcal{O} \\ \frac{1}{2}\text{sym}(\mathcal{A}^{(1)}) \\ \frac{1}{2}\text{sym}(\mathcal{A}^{(2)}) \\ \vdots \\ \frac{1}{2}\text{sym}(\mathcal{A}^{(2)}) \\ \frac{1}{2}\text{sym}(\mathcal{A}^{(1)}) \end{bmatrix} \right), \quad (15)$$

where the $(M-1)$ -order tensors are obtain from holding the M -th index at l , respectively, for $l = 1, 2, \dots, N$. When applied to the degree tensor and the Laplacian tensor, we obtain \mathcal{D}_s and \mathcal{L}_s , respectively. Notice that this operation is reversible. The symmetric hypergraph shift is then given by

$$\vec{y}_s = \mathcal{F}_s * \vec{x}_s. \quad (16)$$

In Section IV-A, we include further discussions on the effects of the symmetrization operation given by the connection of the t-product and the Fourier transform.

In GSP, the graph signal is defined as an N -length vector $\mathbf{x} = [x_1, \dots, x_N]$ where each signal element is related to one node in the hypergraph. In the proposed framework, given that the shifting operation is defined by the t-product and the shifting operator is a tensor of dimension $N \times N \times N_s \times \dots \times N_s$, the hypergraph signal \vec{x}_s and its one-time shifted signal \vec{y}_s should both be tensors of size $N \times 1 \times N_s \times \dots \times N_s$ to have consistent operations. Thus, we now relate a tubal scalar $(\mathbf{x}_s)_{i,1} \in \mathbb{R}^{1 \times 1 \times N_s \times \dots \times N_s}$, $1 \leq i \leq N$, to each node in the hypergraph as shown in Fig. 5 (top-right). This setting opens up the possibilities for different hypergraph signal configurations. Two different of many possibilities are presented next.

Definition 8 (Hypergraph Signal from a Set of Signals): Let $\mathbf{X} = [x_1, x_2, \dots, x_L] \in \mathbb{R}^{N \times L}$ be a set of one dimensional signals on the hypergraph. One can consider assigning to each tubal scalar $(\mathbf{x}_s)_{i,1} \in \mathbb{R}^{1 \times 1 \times N_s \times \dots \times N_s}$, the set of signals corresponding to that node, i.e. $\{x_{i,1}, \dots, x_{i,L}\}$, $1 \leq i \leq N$. Then a total number of $L \leq N_s^{(M-2)}$ signals can be jointly processed on the hypergraph. This setting can be particularly useful when processing signals that are correlated along the tubes, as it is the case of hyperspectral information. We demonstrate this in

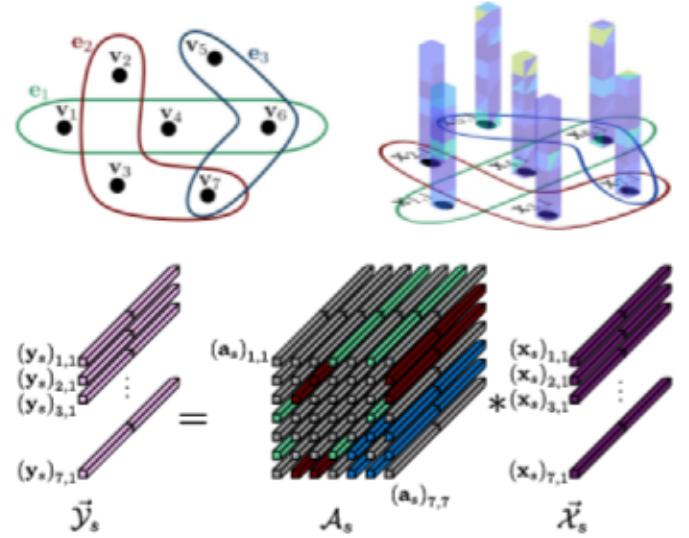


Fig. 5. (top-left) Hypergraph H with set of nodes $V(H) = \{v_1, v_2, \dots, v_7\}$ and set of hyperedges $E(H) = \{e_1, e_2, e_3\}$. (top-right) The hypergraph signal maps a tubal scalar to each node. (bottom) Shifting of the signal \vec{X}_s by the adjacency tensor \mathcal{A}_s . Gray colored tubes are all zero tubal scalars.

our experiments by considering the sparse representation of a hyperspectral point cloud.

Definition 9 (Hypergraph Signal from One Dimensional Signal): For a hypergraph with N nodes and $m.c.e(H) = M$, in [9], an alternative form of hypergraph signal is defined as an $(M-1)$ th-order N -dimensional tensor \mathcal{X} computed as the outer product of an original signal in the hypergraph $\mathbf{x} = [x_1, \dots, x_N] \in \mathbb{R}^N$, i.e. $\mathcal{X} = \mathbf{x} \circ \dots \circ \mathbf{x} \in \mathbb{R}^{N^{(M-1)}}$ where each entry position $\mathcal{X}(i_1, i_2, \dots, i_{M-1})$ equals the product $x_{i_1} x_{i_2} \dots x_{i_{M-1}}$. This definition of hypergraph signal can be easily adapted to the proposed t-product framework by expanding on a new second dimension, $\vec{\mathcal{X}} = \text{expand}(\mathcal{X})$ where $\vec{\mathcal{X}}$ is an M th-order tensor with dimensions $N \times 1 \times N \times \dots \times N$ and by computing its symmetric version as $\vec{X}_s = \text{sym}(\vec{\mathcal{X}})$ such that $\vec{X}_s \in \mathbb{R}^{N \times 1 \times N_s \times \dots \times N_s}$. Notice that as in the CP-based HGSP framework [9], the hypergraph signal \vec{X}_s is just another representation of an original one dimensional signal \mathbf{x} that aims at reflecting its properties in different dimensions. For instance, for a hypergraph with $M = 3$, the hypergraph signal highlights the properties of the 2-D signal components $x_i x_j$.

As an example, let the adjacency tensor be the shifting operator, i.e. $\mathcal{F}_s = \mathcal{A}_s \in \mathbb{R}^{N \times N \times N_s}$, and consider the 3-uniform hypergraph in Fig. 5. The shifted signal in node v_7 is then computed as

$$\begin{aligned} (\mathbf{x}_s)_{7,1} &= (\mathbf{a}_s)_{7,2} * (\mathbf{x}_s)_{2,1} + (\mathbf{a}_s)_{7,3} * (\mathbf{x}_s)_{3,1} \\ &\quad + (\mathbf{a}_s)_{7,5} * (\mathbf{x}_s)_{5,1} + (\mathbf{a}_s)_{7,6} * (\mathbf{x}_s)_{6,1}, \end{aligned} \quad (17)$$

where $(\mathbf{a}_s)_{7,2}$ and $(\mathbf{a}_s)_{7,3}$ are tubal scalars of the symmetrized adjacency tensor \mathcal{A}_s that represent the hyperedge $e_2 = \{v_2, v_3, v_7\}$ and $(\mathbf{a}_s)_{7,5}$ and $(\mathbf{a}_s)_{7,6}$ represent the hyperedge $e_3 = \{v_5, v_6, v_7\}$, which are the only two hyperedges that contain the node v_7 . Note that the entries of the adjacency tensor are very sparse, the tubal scalar $(\mathbf{a}_s)_{7,2}$, for instance, has only

two elements different from zero: $(\mathbf{a}_s)_{7,2}^{(4)}$ which corresponds to the entry $a_{723} \in \mathcal{A}$ and its reflection $(\mathbf{a}_s)_{7,2}^{(13)}$ which is product of the symmetrization operation. Given that $(\mathbf{a}_s)_{7,2}^{(4)} = (\mathbf{a}_s)_{7,2}^{(13)} = a_{723}/2$, we will use, for simplicity, the entries of the adjacency tensor $a_{i,j,k} \in \mathcal{A}$ in this example. Then, for any hypergraph signal, the l -th frontal slice of the i -th tubal scalar $(\mathbf{y}_s)_{i,1}$ can be computed as:

$$(\mathbf{y}_s)_{i,1}^{(l)} = \sum_{\{\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k\} \in \mathbf{E}(\mathbf{H})} \frac{a_{i,j,k}}{2} \left(\mathbf{x}_{j,1}^{[k+l]N_s} + \mathbf{x}_{j,1}^{[N_3-k+l]N_s} \right) \quad (18)$$

for $1 \leq l \leq N_s$, $1 \leq i, j, k \leq N$. The operator $[a]_{N_s} = a \bmod N_s$.

Now, on one hand, one could consider a hypergraph signal obtained from a set of signals \mathbf{X} as in Definition 8, then from 18, the frontal slices of the shifted signal in node \mathbf{v}_7 are given by

$$\begin{aligned} (\mathbf{y}_s)_{7,1}^{(1)} &= \frac{a_{723}}{2} \left(\mathbf{x}_{2,1}^{(4)} + \mathbf{x}_{2,1}^{(13)} \right) + \frac{a_{732}}{2} \left(\mathbf{x}_{3,1}^{(3)} + \mathbf{x}_{3,1}^{(14)} \right) \\ &+ \frac{a_{756}}{2} \left(\mathbf{x}_{5,1}^{(7)} + \mathbf{x}_{5,1}^{(10)} \right) + \frac{a_{765}}{2} \left(\mathbf{x}_{6,1}^{(6)} + \mathbf{x}_{6,1}^{(11)} \right), \\ &\vdots & \vdots \\ (\mathbf{y}_s)_{7,1}^{(15)} &= \frac{a_{723}}{2} \left(\mathbf{x}_{2,1}^{(3)} + \mathbf{x}_{2,1}^{(12)} \right) + \frac{a_{732}}{2} \left(\mathbf{x}_{3,1}^{(2)} + \mathbf{x}_{3,1}^{(13)} \right) \\ &+ \frac{a_{756}}{2} \left(\mathbf{x}_{5,1}^{(6)} + \mathbf{x}_{5,1}^{(9)} \right) + \frac{a_{765}}{2} \left(\mathbf{x}_{6,1}^{(5)} + \mathbf{x}_{6,1}^{(10)} \right), \end{aligned}$$

which represents a linear combination of the signal tubes at neighboring nodes. Note that there is a sequential shift along the third dimension for each frontal slice of the shifted signal. This sequential shift along the third dimension is beneficial to signals that are naturally correlated along the third dimension, as in the case of hyperspectral information. Then, the shifting across the third dimension can exploit the correlation among different channels and benefit applications such as compression and denoising.

On the other hand, one could consider a hypergraph signal from one dimensional signal \mathbf{x} as in Definition 9, then from 18, the frontal slices of the shifted signal in node \mathbf{v}_7 are

$$\begin{aligned} (\mathbf{y}_s)_{7,1}^{(1)} &= 1/2(a_{723}x_2x_3 + a_{732}x_3x_2 \\ &+ a_{765}x_6x_5 + a_{756}x_5x_6), \\ (\mathbf{y}_s)_{7,1}^{(2)} &= 1/4(a_{723}(x_2x_2 + x_2x_4) + a_{732}(x_3x_1 + x_3x_3) \\ &+ a_{756}(x_5x_5 + x_5x_7) + a_{765}(x_6x_4 + x_6x_6)), \\ &\vdots \\ (\mathbf{y}_s)_{7,1}^{(8)} &= 1/4(a_{723}(x_2x_5 + x_2x_4) + a_{732}(x_3x_5 + x_3x_6) \\ &+ a_{756}(x_5x_1 + x_5x_2) + a_{765}(x_6x_2 + x_6x_3)), \end{aligned} \quad (19)$$

where $x_i \in \mathbf{x}$. Note that the first frontal slice $(\mathbf{y}_s)_{7,1}^{(1)}$ correspond to the shifted signal proposed in the CP-based HGSP

Algorithm 1: t-eigendecomposition for a 3-order tensor [42].

Input: $\mathcal{F}_s \in \mathbb{R}^{N \times N \times N_s}$
Output: $\Lambda \in \mathbb{R}^{N \times N \times N_s}$, $\mathcal{V} \in \mathbb{R}^{N \times N \times N_s}$

- 1: $\hat{\mathcal{F}}_s \leftarrow \text{fft}(\mathcal{F}_s, [], 3)$
- 2: **for** $k \leftarrow 1$ to N_s **do**
- 3: $[\mathbf{V}, \mathbf{D}] = \text{eig}(\hat{\mathcal{F}}_s(:, :, k))$
- 4: $\hat{\mathcal{V}}(:, :, k) \leftarrow \mathbf{V}$; $\hat{\Lambda}(:, :, k) \leftarrow \mathbf{D}$;
- 5: **end for**
- 6: $\Lambda \leftarrow \text{ifft}(\hat{\Lambda}, [], 3)$
- 7: $\mathcal{V} \leftarrow \text{ifft}(\hat{\mathcal{V}}, [], 3)$

approach [9] but divided by two. Different from [9], the frontal slices 2 through $N + 1$ consider the cross-product of the signals at nodes connected to node 7 and the rest of the nodes in the graph.

IV. HYPERGRAPH SPECTRUM SPACE

Central to HGSP is the Fourier transform. In our formulation, the generalization comes directly from the tensor eigen-decomposition (t-eigendecomposition). Given the symmetric shifting operator $\mathcal{F}_s \in \mathbb{R}^{N \times N \times N_s}$, the t-eigendecomposition (Algorithm 1) is determined by

$$\mathcal{F}_s = \mathcal{V} * \Lambda * \mathcal{V}^T, \quad (20)$$

where $\mathcal{V} = [\vec{\mathcal{V}}_1, \dots, \vec{\mathcal{V}}_N] \in \mathbb{R}^{N \times N \times N_s}$ is an orthogonal tensor, $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N) \in \mathbb{R}^{N \times N \times N_s}$ is an f-diagonal tensor whose frontal slices are diagonal matrices, and $(\lambda_j \in \mathbb{R}^{1 \times 1 \times N_s}, \vec{\mathcal{V}}_j \in \mathbb{R}^{N_1 \times 1 \times N_3})$ corresponds to the j -th eigen-pair of eigen-tuple and its corresponding eigen-matrix. The hypergraph Fourier transform of a hypergraph signal $\vec{\mathcal{X}}_s \in \mathbb{R}^{N \times 1 \times N_s}$ (t-HGFT) is then $\vec{\mathcal{X}}_{F_s} = \vec{\mathcal{V}} * \vec{\mathcal{X}}_s$, with the inverse hypergraph Fourier transform (t-IHGFT) given by $\vec{\mathcal{X}}_s = \mathcal{V} * \vec{\mathcal{X}}_{F_s}$. Since the tensor \mathcal{V} forms an orthonormal set with cardinality N , i.e., $\vec{\mathcal{V}} * \mathcal{V} = \mathcal{V} * \vec{\mathcal{V}} = \mathcal{I}_{NNN_s}$, perfect Fourier representation and recovery of the signal is achieved. See Fig. 3(b).

A. Hypergraph Frequency

Given the above definition of the hypergraph Fourier transform, the next step is to determine a notion of frequency, which is key for filtering. In GSP, the notion of frequency is provided by the eigenvalues and eigenvectors of the representing matrix \mathbf{F} ordered based on the total variation (TV) [44]. The total variation and hence the frequency describes the rate of signal changes among neighbors. For t-HGSP, the total variation of the hypergraph Fourier basis can also be used to define a notion of frequency. To this end, we first define the total variation on hypergraphs around the concept of the hypergraph shifting and use this definition to determine an ordering of the eigen-tuples of the adjacency tensor, \mathcal{A}_s , that provides a notion of frequency.

Definition 10 (Total Variation on a Hypergraph): Let $\mathcal{F}_s = \mathcal{A}_s$. The total variation of a hypergraph signal $\vec{\mathcal{X}}_s$ on \mathbf{H} is defined as the similarity measure between the hypergraph signal $\vec{\mathcal{X}}_s$ and

its shifted version $\vec{\mathcal{Y}}_s$ computed as

$$\text{TV}_{\mathbf{H}}(\vec{\mathcal{X}}_s) = \|\vec{\mathcal{X}}_s - \vec{\mathcal{Y}}_s\| = \|\vec{\mathcal{X}}_s - \mathcal{F}_s^{\text{norm}} * \vec{\mathcal{X}}_s\|,$$

$\mathcal{F}_s^{\text{norm}} = \mathcal{V} * \Lambda^{\text{norm}} * \mathcal{V}^{-1}$ where Λ^{norm} is a normalized diagonal tensor. The k -th frontal slice of Λ^{norm} is computed in the Fourier domain as $(\hat{\Lambda}^{\text{norm}})^{(k)} = |\lambda_{\max}^{(k)}|^{-1} \hat{\Lambda}^{(k)}$. The normalization ensures numerical stability when filtering as it prevents excessive scaling of the shifted signal [44]. The norm $\|\cdot\|$ is computed according to the t-product Definition 4.

Since the TV describes how oscillatory the signal is on the hypergraph, it leads to the concept of low and high frequencies on the hypergraph Fourier spectrum, \mathcal{V} , as shown next.

Theorem 1: Let $\mathcal{F}_s = \mathcal{A}_s$ and λ_i and λ_j be two eigen-tuples of \mathcal{F}_s with corresponding eigen matrices $\vec{\mathcal{V}}_i$ and $\vec{\mathcal{V}}_j$. Given that \mathcal{F}_s is symmetric, it has real eigen-tuples. If the eigen-tuples are ordered as $\lambda_i > \lambda_j$, then the total variations of their eigenvectors satisfy $\text{TV}_{\mathbf{H}}(\vec{\mathcal{V}}_i) < \text{TV}_{\mathbf{H}}(\vec{\mathcal{V}}_j)$.

Proof: See Appendix A.

Theorem 1 provides an ordering of frequency according to the eigen-tuples λ_j 's. As a result, the hypergraph Fourier basis of the adjacency tensor \mathcal{A}_s which are the eigen-pairs $(\lambda_j, \vec{\mathcal{V}}_j)$ are ordered from lowest to highest frequency if $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_j \geq \dots \geq \lambda_N$. Recall that tubal scalars are ordered according to Definition 3.

In parallel to GSP theory, we can also define the Laplacian-based total variation and provide an ordering for the hypergraph Fourier basis of the Laplacian tensor \mathcal{L}_s . Hence, we define the Laplacian-based total variation on a hypergraph as

$$\text{TV}_{\mathcal{L}}(\vec{\mathcal{X}}_s) = \vec{\mathcal{X}}_s * \mathcal{L}_s * \vec{\mathcal{X}}_s. \quad (21)$$

Then, the total variation of a Fourier basis vector $\vec{\mathcal{V}}_j$ is

$$\text{TV}_{\mathcal{L}}(\vec{\mathcal{V}}_j) = \lambda_j. \quad (22)$$

In contrast to the Fourier basis obtained from the adjacency tensor, the eigenvectors of the Laplacian shifting operator are ordered from lowest frequency to highest as $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_j \leq \dots \leq \lambda_N$. In simple graphs, the eigenspace of the graph Laplacian $\mathbf{L} = \mathbf{V}\mathbf{A}\mathbf{V}^T$ provides a similar notion of frequency as in classical Fourier analysis. The graph Laplacian eigenvectors associated with smaller eigenvalues vary slowly across the graph and the eigenvectors associated with larger eigenvalues oscillate more rapidly and are more likely to have dissimilar values on vertices connected by an edge [5]. For the hypergraph Fourier transform defined here, in Fig. 6, we visually compare the variation among three of the column vectors in $\hat{\mathcal{V}}^{(1)}$ which is the first frontal slice of $\hat{\mathcal{V}}$. Note that there is higher variation among neighboring nodes as the value of the eigenvalue increases. Given the notion of frequency, we then introduce the definition of the bandlimited signals on a hypergraph as follows.

Definition 11 (Band-limited Signal): A hypergraph signal $\vec{\mathcal{X}}_s \in \mathbb{R}^{N \times 1 \times N_s}$ is a K-band-limited if its t-HGFT, $\vec{\mathcal{X}}_{F_s}$, has tubal scalars $(\mathbf{x}_{F_s})_{l,1} = 0$ for all $l > K$ where $K \in 1, 2, \dots, N$. The smallest K is called the bandwidth of $\vec{\mathcal{X}}_s$.

Note that hypergraph band-limited signals are not required to be low-pass or smooth, in which the eigen-tuples are sorted

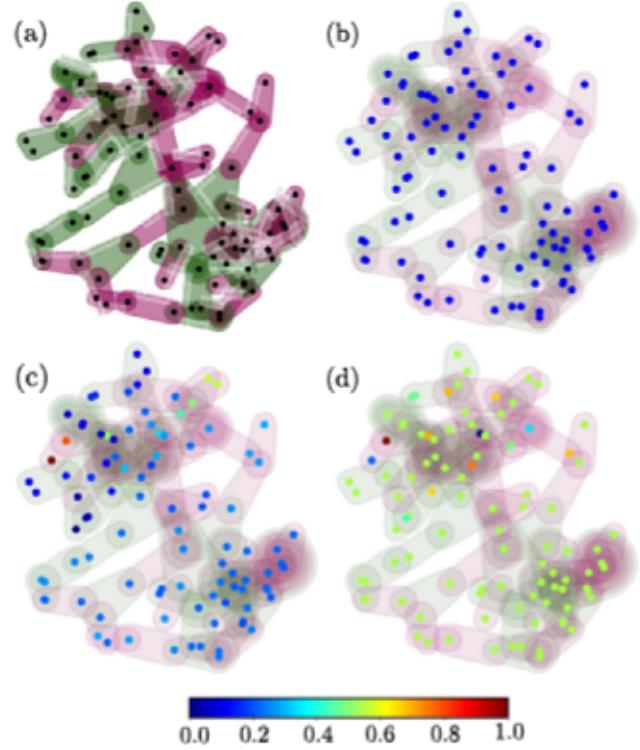


Fig. 6. (a) Hypergraph \mathbf{H} : subset of the DBLP coauthorship network; and, 1-th frontal slice of three different eigen-matrices in the Fourier domain: (b) $\hat{\mathcal{V}}_1^{(1)}$, (c) $\hat{\mathcal{V}}_{10}^{(1)}$, and (d) $\hat{\mathcal{V}}_{59}^{(1)}$ of the hypergraph Laplacian \mathcal{L}_s of \mathbf{H} ; ordered from lowest to highest frequency. The higher the frequency is readily the higher variation of the eigenvector signal values at neighboring nodes. Hyperedges have different cardinality: (green) $|\mathbf{e}| = 3$ and (purple) $|\mathbf{e}| = 2$.

from low to high frequency as explained before. Instead, the band-limited definition is equivalent to restricting the number of nonzero tubal scalars in the hypergraph Fourier spectrum $\vec{\mathcal{X}}_{F_s}$ [29] with no specific ordering on the eigen-tuples.

V. DISCUSSIONS AND INTERPRETATIONS

In this section, we discuss the benefits of using t-product as the core operation for the proposed HGSP framework. We also establish the connection and advantages over classical GSP frameworks, focusing on the aspects that help to interpret the notion of hypergraph frequency.

A. Interpretation of Hypergraph Spectrum Space

To analyze the hypergraph spectrum space we consider not only the connection of HGSP with DSP and GSP but also the connection of the t-product with the Fourier transform.

1) Connection of the T-Product With the Fourier Transform: To better understand the hypergraph spectrum of a signal, we consider the connection of the t-product with the discrete Fourier transform (DFT) [39]. As observed in Algorithm 1, the t-eigendecomposition is efficiently computed by traditional matrix eigendecomposition of the frontal slices of the shifting operator \mathcal{F}_s in the Fourier domain $\hat{\mathcal{F}}_s$. Note that, as depicted in Fig. 7, a better interpretation of the hypergraph spectrum

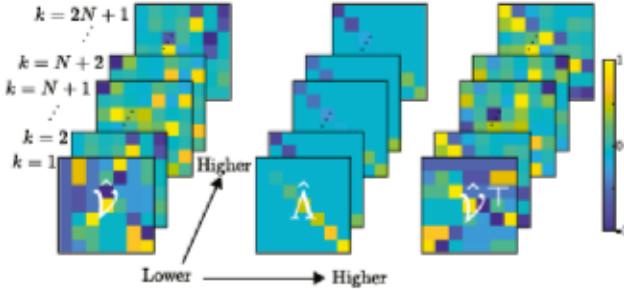


Fig. 7. The hypergraph spectrum space computed from the Laplacian tensor \mathcal{L}_s of the hypergraph in Fig. 5(top-left) is analyzed here. Better interpretability is achieved by considering the connection of the t-product with the DFT [39], which also provides a notion of frequency along the third dimension. Then, each frontal slice of the t-eigendecomposition in the Fourier domain depicted in this figure is efficiently computed by traditional matrix-decomposition $\hat{\mathcal{L}}_s^{(k)} = \hat{\mathcal{V}}^{(k)} \hat{\Lambda}^{(k)} \hat{\mathcal{V}}^{\top(k)}$. Then, for each frontal slice, the notion of lower to higher frequency is provided by the increasing ordering of the eigenvalues as explained before in Section IV-A, and a notion of lower to higher frequency along the third dimension is given by the DFT of the tubal scalars.

can be obtained by analyzing the t-eigendecomposition in the Fourier domain. Then, for each frontal slice, the notion of frequency is provided by ordering the eigenvalues as explained in Section IV-A. Additionally, a notion of frequency along the third dimension is provided by the DFT of the tubal scalars.

Note that one of the important properties of the DFT is that when the signal is real and even, which is the case of the tubal scalars in the symmetrized tensors, its Fourier spectrum is also real and even. This can also be observed in Fig. 7 where the DC component is represented by $k = 1$, and the rest of frequencies, from lowest to highest, are given by the pair of frontal slices $k = p + 1 \Leftrightarrow k = 2(N + 1) - p$ for $p = 1, \dots, N$, with $k = N + 1 \Leftrightarrow k = N + 2$ being the highest frequency. Here, we see that by the symmetrization of the shifting operator, we guarantee that its Fourier domain representation is real and even which also leads to an eigendecomposition with real eigenvalues, which is characteristic of undirected graphs and hence a desired property for undirected hypergraphs.

2) *Connection With DSP and GSP*: To obtain an intuitive interpretation on the hypergraph frequency, we consider its relationship with DSP and GSP. In DSP, the discrete Fourier transform (DFT) of a signal $s = \{s_n : n = 0, 1, \dots, N - 1\}$ is $\hat{s} = \{\hat{s}_k : k = 0, 1, \dots, N - 1\}$ where each Fourier coefficient is given by $\hat{s}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} s_n e^{-j2\pi kn/N}$ [6]. The discrete frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \dots, N - 1$ are related to the degree of variation of the spectral components. To observe this, we consider the definition of the total variation in DSP:

$$\text{TV}(s) = \sum_{n=0}^{N-1} |s_n - s_{[n-1]_N}|, \quad (23)$$

$$= \|s - \mathbf{A}_c s\|_1, \quad (24)$$

where $s_{[n-1]_N}$ is the one-time shifted signal and this shifting operation can be represented by multiplying by the circulant

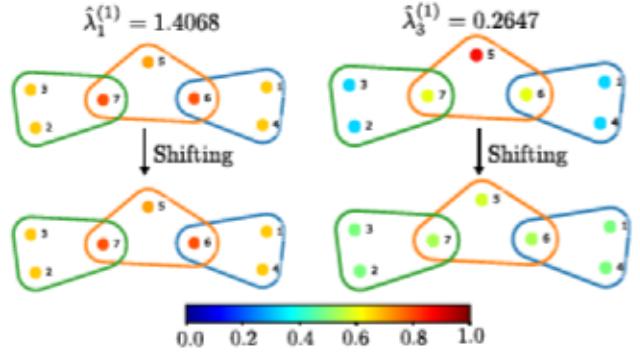


Fig. 8. For the hypergraph example depicted in Fig. 5(top-left), we analyze the total variation of the hypergraph spectrum provided by the adjacency tensor \mathcal{A}_s by considering the connection of the t-product with the Fourier domain. (top-left) 1st frontal slice of the 1st eigen-tensor in the Fourier domain $\hat{\mathcal{V}}_1^{(1)}$ and (top-right) 1st frontal slice of the 3rd eigen-tensor in the Fourier domain $\hat{\mathcal{V}}_3^{(1)}$ and (bottom) their respective shifted representations.

matrix:

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad (25)$$

which is diagonalized by the DFT. Hence, the eigenvalues of \mathbf{A}_c are $\lambda_n = e^{-j2\pi n/N}$, $n = 0, \dots, N - 1$ and the eigenvectors \mathbf{v}_n , $n = 0, \dots, N - 1$ are the columns of the inverse of the discrete Fourier matrix DFT_N^{-1} . Then, the total variation of each eigenvector \mathbf{v}_n , computed as

$$\text{TV}(\mathbf{v}_n) = |1 - e^{-j2\pi n/N}|, \quad (26)$$

induces an ordering from lowest to highest frequencies on the eigenvalues as $\lambda_0, \lambda_1, \lambda_{N-1}, \lambda_2, \lambda_{N-2}, \dots$, with the lowest frequency corresponding to λ_0 and the highest frequency corresponding to $\lambda_{N/2}$ for even N or $\lambda_{(N\pm1)/2}$ for odd N , which is the conventional frequency ordering in DSP [45]. Hence, higher frequency components change faster over time, which implies larger total variation.

Note that, the shift matrix \mathbf{A}_c can be seen as the adjacency matrix of a cycle line graph [6]. Thus, we now consider the total variation and frequency in GSP, replacing \mathbf{A}_c by the normalized shifting operator of any graph as

$$\text{TV}(\mathbf{v}_n) = \|\mathbf{s} - \mathbf{F}^{\text{norm}} \mathbf{v}_n\|_1, \quad (27)$$

where $\mathbf{F}^{\text{norm}} = 1/\lambda_{\max} |\mathbf{F}|$. Then, λ_m is larger than graph frequency λ_l if the $\text{TV}(\mathbf{v}_m) > \text{TV}(\mathbf{v}_l)$ since larger variation indicates faster change in the signal values between neighboring nodes on the graph, which represents higher frequencies. For undirected graphs, the eigenvalues are real numbers, and the frequency decreases with the increase of the increasing the eigenvalues as demonstrated for the general case of hypergraphs in Section IV-A. If the graph is undirected, the eigenvalues are complex and the frequency changes following a similar pattern as explained before for DSP.

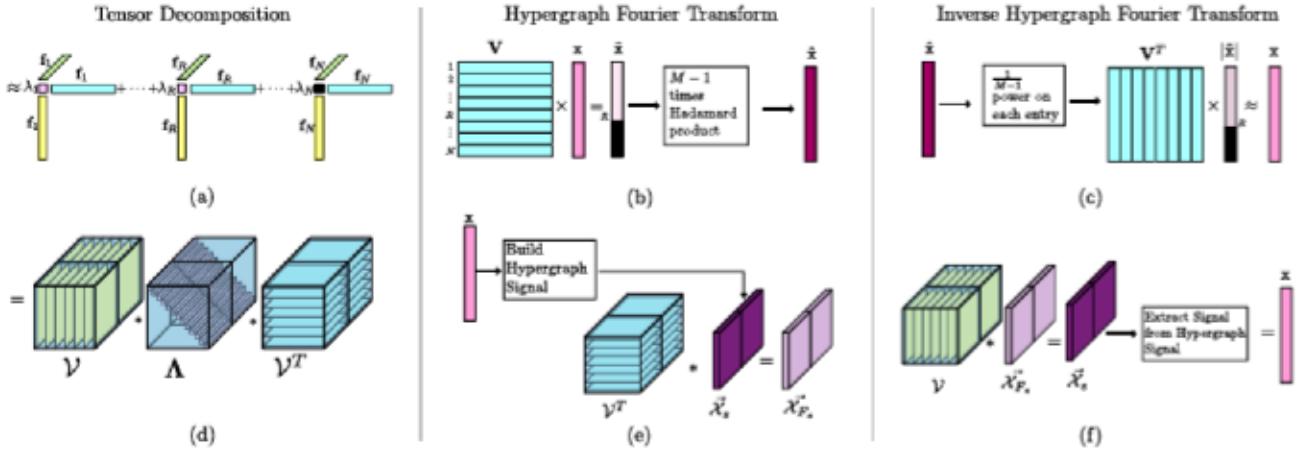


Fig. 9. Comparison of the core elements of (top) the CP-HGSP framework [9] and (bottom) the proposed t-HGSP framework. (a) Orthogonal Symmetric CP decomposition of \mathcal{F} , (b) CP-HGFT, and (c) CP-IHGFT; (d) t-eigendecomposition of \mathcal{F}_s , (e) t-HGFT, and (f) t-IHGFT. Black elements represent zero values.

Now, we consider the generalization to HGSP by replacing the shifting operator by an M -th order tensor \mathcal{F}_s . Then, the generalized TV on hypergraphs in (21) measures the variation on the eigen-tensors components as

$$\text{TV}_H(\vec{\mathcal{V}}_j) = \|\vec{\mathcal{V}}_j - \mathcal{F}_s^{\text{norm}} * \vec{\mathcal{V}}_j\|,$$

where $\mathcal{F}_s^{\text{norm}} = \mathcal{V} * \Lambda^{\text{norm}} * \mathcal{V}^T$. As explained before, given that we consider undirected hypergraphs and use symmetrized shifting operators, the eigen-pairs are real in both the spatial domain $\lambda_j, \vec{\mathcal{V}}_j$ and the Fourier domain $\hat{\lambda}_j, \vec{\mathcal{V}}_j$. Then, as a generalization of GSP and DSP, λ_m is larger than hypergraph frequency λ_1 if the $\text{TV}(\vec{\mathcal{V}}_m) > \text{TV}(\vec{\mathcal{V}}_1)$ since larger variation indicates faster changes in the signal values between neighboring nodes on the hypergraph, which represents higher frequencies. To illustrate this, consider the hypergraph example in Fig. 8 where, clearly, there is higher variation between the eigen-tensor $\vec{\mathcal{V}}_j$ and its shifted version $\mathcal{F}_s^{\text{norm}} * \vec{\mathcal{V}}_j$ as the eigenvalues increase. Note that, we analyze this in the Fourier domain given that the ordering of the eigen-tuples (Definition 3) is better analyzed by considering the connection of the t-product with the Fourier transform.

B. Connections to Other Frameworks

1) *Graph Signal Processing*: HGSP aims at being a more general framework than traditional GSP, enabling the processing of signals in high-dimensional graphs that capture polyadic relationships among the nodes. Thus, the proposed HGSP framework is a generalization of traditional GSP which is a special case of HGSP with $M = 2$. Consider this particular case ($M = 2$) in the core elements of the proposed framework:

- *Algebraic descriptor*: GSP considers the case in which each edge connects exactly two nodes while HGSP considers hyperedges that can connect more than two nodes. A graph is a uniform hypergraph with maximum edge cardinality $M = m.c.e(\mathbf{H}) = 2$ represented by an 2nd-order adjacency tensor $\mathcal{A} \in \mathbb{R}^{N \times N}$ with entries $a_{p_1, p_2} = a_{p_2, p_1} = 1$ if the node \mathbf{v}_{p_1} is connected to node \mathbf{v}_{p_2} and zero otherwise. Note that the symmetrization operation defined in

Section III is applied for $M > 2$ which excludes the case of simple graphs given that they are already symmetric according to Definition 2. Thus, this 2nd-order adjacency tensor $\mathcal{A} \in \mathbb{R}^{N \times N}$ is the binary adjacency traditionally used at the core of GSP.

- *Signal Shifting*: Consider $M = 2$ in the HGSP signals in Definition 9 where the $M - 1$ outer product of the one-dimensional signal is itself. Similarly, in Definition 8, only one one-dimensional signal is processed by the hypergraph when $M = 2$. As a consequence, the graph shifting is also a special case of hypergraph shifting when $M = 2$.
- *Spectral Properties*: the t-eigendecomposition [42], being a generalization of traditional matrix algebra, reduces to traditional matrix eigendecomposition when $M = 2$. Hence, the t-HGFT is the same as the GFT when $M = 2$.

HGSP being a generalization of GSP not only holds the same benefits as the traditional GSP framework but also provides additional advantages. First, the adjacency tensor can, without ambiguities, encode high-order relationships that are widespread in many real-word applications [1], [13], [14], [15]. Second, the high-order shifting operation can either jointly process multiple signals (Definition 8) or consider cross-node relationships (Definition 9), modeling the joint effects of nodes within a hyperedge.

2) *Hypergraph Signal Processing*: Recently, the CP tensor decomposition was used to define a hypergraph signal processing framework [9]. This section compares the core elements of the CP-based HGSP framework [9] with the ones in our framework and states the benefits of using t-product tensor decompositions for HGSP. As shown in Fig 9(a), the HGSP framework in [9] uses the orthogonal-CP decomposition of the shifting operator \mathcal{F} computed as

$$\mathcal{F} \approx \sum_{r=1}^R \lambda_r \underbrace{\mathbf{f}_r \circ \dots \circ \mathbf{f}_r}_{M-\text{times}}, \quad (28)$$

to define the matrix $\mathbf{V} = [\mathbf{f}_1 \dots \mathbf{f}_r \dots \mathbf{f}_R]^T$ which is used to compute the hypergraph Fourier transform. Remember that

\mathcal{F} represents either the Adjacency tensor \mathcal{A} or the Laplacian tensor \mathcal{L} which are super-symmetric tensors. As depicted in Fig. 9(b), [9] defines the spectrum of the signal $\mathbf{x} \in \mathbb{R}^N$ on the hypergraph as $\tilde{\mathbf{x}} = \mathbf{V}\mathbf{x}$ where $\mathbf{V} = [\mathbf{f}_1, \dots, \mathbf{f}_N]^\top$. Then for the hypergraph signal defined as $\mathbf{x}^{[M-1]} = \underbrace{\mathbf{x} \odot \dots \odot \mathbf{x}}_{M-1}$,

the hypergraph Fourier space $\tilde{\mathbf{x}}$ can be computed as $M-1$ times the Hadamard product of $\tilde{\mathbf{x}}$, that is $\tilde{\mathbf{x}} = \underbrace{\tilde{\mathbf{x}} \odot \dots \odot \tilde{\mathbf{x}}}_{M-1} =$

$[(\mathbf{f}_1^\top \mathbf{x})^{M-1}, \dots, (\mathbf{f}_N^\top \mathbf{x})^{M-1}]^\top$. The drawback of this approach is that tensors in general do not have an exact CP symmetric orthogonal decomposition [37] as shown in the example in Fig. 3(a), and finding the rank R of a tensor is NP-hard. Furthermore, since usually the CP-rank $R < N$, the authors in [9] add additional vectors \mathbf{f}_r by using zero coefficients λ_r for $r = R+1, \dots, N$. These $N-R$ additional vectors \mathbf{f}_r should meet the conditions of: (1) orthogonality $\mathbf{f}_r \perp \mathbf{f}_i$ for $r \neq i$, (2) normalization with $|\mathbf{f}_r| = 1$, and (3) $\mathbf{f}_r^\top \mathbf{x} = 0, \forall r > R$. However, \mathbf{f}_r vectors satisfying these conditions can only be added when the signal is band-limited with bandwidth $W \leq R$. Thus, as illustrated in Fig. 9(c) and in the example in Fig. 3(a), since \mathbf{x} lives in \mathbb{R}^N , the hypergraph Fourier transform defined in [9] maps the signal to a subspace of \mathbb{R}^N when $R < N$. Hence, perfect recovery on the inverse HGFT of $\mathbf{x} \in \mathbb{R}^N$ is not guaranteed unless the signal is band-limited with bandwidth $W \leq R$. Note that one might consider conditions different from [9] in order to obtain a loss-free HGFT when using the CP symmetric orthogonal decomposition. On the one hand, one could set the rank to N , which would likely lead to higher errors on the tensor decomposition since the rank is not optimal, but a loss-free HGFT would be obtained. On the other hand, one might ignore the third condition on the additional \mathbf{f}_r vectors, optimize the rank to find the best tensor approximation, and then add $N-R$ additional vectors \mathbf{f}_r associated to zero λ_r coefficients which are not necessarily orthogonal to the signal. For the latter, the notion of frequency would be lost on these additional bases, given that they are all assigned to zero eigenvalues.

In the framework introduced here, the t-eigendecomposition always provides an exact tensor factorization of the shifting operator. Particularly, as shown in Fig. 9(d), when this tensor is symmetrized \mathcal{F}_s , the t-eigendecomposition is composed by an orthogonal tensor of eigen-matrices \mathcal{V} and an f-diagonal tensor of eigen-tuples \mathbf{A} . This representation allows for a loss-free generalization of the basic elements in a HGSP framework: the hypergraph Fourier transform (t-HGFT), in Fig. 9(e), and the inverse hypergraph Fourier transform (t-iHGFT), in Fig. 9(f). Moreover, the HGFT and iHGFT, defined here, are more robust than those obtained in the CP-based approach, given that any t-product tensor decomposition algorithm could be used and lead to reproducible results. In contrast, the performance of the CP-based approach relies on selecting a suitable tensor decomposition algorithm. Then, the advantages of adopting t-eigendecompositions to define an HGSP framework with respect to previous approaches are compelling — (1) t-product decompositions preserve the intrinsic structures of tensors; (2) the high-dimensional nature of signals is preserved; (3) the orthogonal eigenbasis spans the whole signal space which allows

for loss-free Fourier transforms [42] and the symmetric properties of the tensors together with the Fourier domain computation of t-products allow for computationally efficient calculations.

In the following sections, essential tools in DSP are formulated within the context of the proposed t-HGSP framework and the aforementioned advantages are demonstrated through simulations on different types of data.

VI. HYPERGRAPH FILTERS

Filtering is an essential tool used in signal processing applications as measurement noise is always present [46]. Filtering in standard DSP amplifies or attenuates the contributions of each of signal's frequency components as $\hat{y}(f) = \hat{q}(f)\hat{x}(f)$ where $\hat{q}(f)$ is the transfer function of the filter. Multiplication in the Fourier domain corresponds to convolution in the time domain, i.e., $y(t) = (q \star x)(t)$ [5]. Notably, these fundamental concepts have a natural parallel formulation in the proposed hypergraph Fourier transform — frequency filtering can be directly generalized as $\mathbf{y}_{F_s}(\lambda_l) = \mathbf{q}_{F_s}(\lambda_l) * \mathbf{x}_{F_s}(\lambda_l)$, where $\mathbf{y}_{F_s}(\lambda_l), \mathbf{x}_{F_s}(\lambda_l), \mathbf{q}_{F_s}(\lambda_l) \in \mathbb{R}^{1 \times 1 \times 1+2^s N}$ are, respectively, the tubal scalars at frequency λ_l of the output signal, \vec{y}_s , the input signal, \vec{x}_s , and the filter response, Q_s , in the frequency domain. When taking the inverse Fourier transform of \vec{y}_s , the tubal scalars of \vec{y}_s are given by

$$(\mathbf{y}_s)_{j,1} = \sum_{l=1}^N \mathbf{v}_{j,l} * \mathbf{q}_{F_s}(\lambda_l) * \mathbf{x}_{F_s}(\lambda_l), \quad (29)$$

which can be written in tensor-tensor product notation as

$$\vec{y}_s = Q_s * \vec{x}_s \\ = \mathcal{V} * \begin{bmatrix} \mathbf{q}_{F_s}(\lambda_1) & \dots & 0 \\ & \ddots & \\ 0 & & \mathbf{q}_{F_s}(\lambda_N) \end{bmatrix} * \underbrace{\mathcal{V}^\top * \vec{x}_s}_{\vec{x}_{F_s}}. \quad (30)$$

In GSP, graph spectral filtering is used to implement well-known filtering techniques designed via optimization formulations such as Gaussian smoothing, bilateral filtering, total variation filtering, anisotropic diffusion, and nonlocal means filtering [32]. Many of these filters, in fact, arise as solutions of regularized ill-posed inverse problems such as denoising, inpainting, and super resolution. Hypergraph filters can also be designed via optimization approaches [9]. As an example, let \vec{y}_s be the second moments of the observed noisy signal. To enforce a priori information that the second moments of the clean signal \vec{x}_s are smooth with respect to the underlying hypergraph, we include a regularization term as

$$\operatorname{argmin}_{\vec{x}_s} \|\vec{x}_s - \vec{y}_s\| + \mathcal{D}[\mathbf{d}] * \Phi(\vec{x}_s, \mathcal{F}), \quad (31)$$

where $\mathbf{d} = \gamma \mathbf{e}_1$, and $\mathcal{D}[\mathbf{d}]$ is a diagonal tensor with \mathbf{d} repeated down the diagonal and $\Phi(\vec{x}_s, \mathcal{F})$ is a function that measures the smoothness of the signal \vec{x}_s in the hypergraph. Well-known optimization problems with types of regularizations such as the Tikhonov regularization have been previously formulated and solved for third order tensors in the literature [42]. A

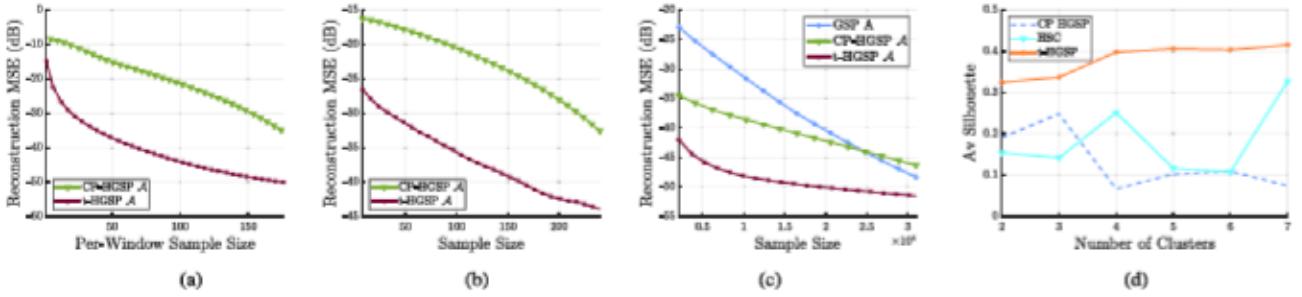


Fig. 10. Reconstruction MSE comparison in the compression of the signal: (a) Cameraman, (b) movie ratings and (c) hyper-spectral point cloud. (d) Performance comparison of hypergraph spectral clustering on the Zoo dataset by measuring the average silhouette when having different number of clusters.

numerical experiment demonstrating this approach is included in Section III of the Supplemental Material.

VII. SAMPLING ON HYPERGRAPHS

Sampling is one of the fundamental concepts in DSP. Standard sampling theory relies on concepts of frequency domain analysis, shift-invariant signals, and band-limitedness. A common problem in networks, usually described by graphs, is to determine which nodes play the most important role. Graph signal sampling then becomes an essential tool for a GSP framework. Traditional sampling theory and spectral graph theory have been combined to generalize Nyquist sampling principles on graphs [29], [47], [48], [49], [50], [51], [52], [53]. Suppose that we want to sample Q tubal scalars from the hypergraph signal $\vec{\mathcal{X}}_s \in \mathbb{R}^{N \times 1 \times N_s}$, then the sampled signal $\vec{\mathcal{X}}_s(S) \in \mathbb{R}^{Q \times 1 \times N_s}$ with $Q < N$ and $S = \{s_1, s_2, \dots, s_Q\} \subset V(\mathbf{H})$ denotes the sequence of sampled nodes. For the sampling of these Q tubal scalars, the sampling operator is defined as a 3rd-order tensor $\Psi \in \mathbb{R}^{Q \times N \times N_s}$ whose first frontal slice is given by

$$\Psi_{i,j}^{(1)} = \begin{cases} 1, & j = s_i \in S, \\ 0, & \text{otherwise.} \end{cases} \quad (32)$$

and the other frontal slices are all zeros. Now, to recover $\vec{\mathcal{X}}_s \in \mathbb{R}^{N \times 1 \times N_s}$, let's define an interpolation operator, $\Phi \in \mathbb{R}^{N \times Q \times N_s}$, such that

$$\vec{\mathcal{X}}_s^* = \Phi * \vec{\mathcal{X}}_s(S), \quad (33)$$

$$= \Phi * \Psi * \vec{\mathcal{X}}_s, \quad (34)$$

where $\vec{\mathcal{X}}_s^* \in \mathbb{R}^{N \times 1 \times N_s}$ approximates $\vec{\mathcal{X}}_s$ either exactly or approximately [44]. For perfect recovery $\Phi * \Psi$ must be equal to the identity tensor which is, in general, not possible since the multi-rank (Definition 7) of $\Phi * \Psi$, $\rho(\Phi * \Psi) \in \mathbb{R}^{1 \times 1 \times N_s}$, has frontal entries $\rho(\Phi * \Psi)^{(k)} \leq Q < N$. However, perfect recovery can be achieved for band-limited signals considering the following property.

Lemma 1: If $\vec{\mathcal{X}}_s$ is a K-band-limited signal then

$$\vec{\mathcal{X}}_s = \mathcal{V}_{[K]} * \left(\vec{\mathcal{X}}_{F_s} \right)_{[K]}, \quad (35)$$

where $\mathcal{V}_{[K]} = [\vec{\mathcal{V}}_1, \dots, \vec{\mathcal{V}}_K] \in \mathbb{R}^{N \times K \times 2N+1}$ and $(\vec{\mathcal{X}}_{F_s})_{[K]} \in \mathbb{R}^{K \times 1 \times 2N+1}$ correspond to the K nonzero elements of the hypergraph signal in the frequency domain $\vec{\mathcal{X}}_{F_s}$.

Proof: See Appendix B.

In this lemma, it is implied that K frequency components carry all the information of the signal; hence the GSP theory on the perfect recovery of band-limited signals [29] can be generalized to hypergraphs by the following theorem.

Theorem 2: Let $\mathcal{V}_{[K]}$ be the tensor formed by K lateral slices of \mathcal{V} , i.e. the orthonormal set $\{\vec{\mathcal{V}}_1, \vec{\mathcal{V}}_2, \dots, \vec{\mathcal{V}}_K\}$, and let the sampling operator $\Psi \in \mathbb{R}^{Q \times N \times N_s}$ satisfy

$$\rho(\Psi * \mathcal{V}_{[K]})^{(k)} = K, \quad (36)$$

for $i = 1, 2, \dots, 2N + 1$. Since $\rho(\Psi * \mathcal{V}_{[K]})^{(k)} \leq \min(Q, K)$, the sample size Q has to be greater than or equal than the bandwidth K . The interpolation operator $\Phi = \mathcal{V}_{[K]} * \mathcal{U} \in \mathbb{R}^{N \times Q \times N_s}$ with $\mathcal{U} * \Psi * \mathcal{V}_{[K]} = \mathcal{I}_{K \times K \times N_s}$, achieves perfect recovery, i.e. $\vec{\mathcal{X}}_s = \Phi * \Psi * \vec{\mathcal{X}}_s$ for any K -bandlimited signal.

Proof: See Appendix C.

As in GSP, from Theorem 2, we observe that an arbitrary sampling operator may not lead to perfect recovery even for band-limited hypergraphs signals [29]. To have perfect recovery, the sampling operator Ψ must be a *qualified sampling operator* by satisfying the multirank condition (36). The optimization of the sampling pattern such that Ψ is *qualified sampling operator* and robust to noise is part of our future work. It is important to note that, for hypergraphs, a sampling operator Ψ that satisfies the multirank condition (36) is not limited to the sampling of tubal scalars as in 32, but instead a *qualified sampling operator* Ψ could be designed such that the sampled signal $\vec{\mathcal{X}}_s(S)$ is formed by the linear combination of the signals in $\vec{\mathcal{X}}_s$. This type of sampling where only a linear combination of the original signal can be obtained is present in many applications such as compressive spectral imaging (CSI) systems which aim at capturing large volumes of spatio-spectral information of a scene of interest from a set of noisy undersampled observations [54], [55].

VIII. EXPERIMENTS

A. Sparse Hypergraph Signal Representation

Sparse signal representations assume that most of the information can be captured by just a few basis functions in an orthonormal basis set [56]. Moments and cumulants are commonly used in the statistical analysis of random variables [57]. However, storing empirical moment tensors can be prohibitively expensive. Here, we demonstrate that the proposed hypergraph Fourier Transform methodology can be successfully applied to obtain a sparse representation of a hypergraph signal carrying empirical moments (Definition 9). To this end, we consider the ‘‘Cameraman’’ image of size 128×128 which is first tiled into non-overlapping blocks of size 16×16 . As in [9], for each block, a hypergraph with a 3rd-order *adjacency tensor*, \mathcal{A} , is obtained by using the image adaptive neighborhood hypergraph (IANH) model [41]. The *hypergraph signal*, $\tilde{\mathcal{X}}_s$, was built from the one dimensional grayscale signal according to Definition 9, capturing its 2-D properties. The sparse representation of the hypergraph signal is then obtained by keeping only the largest C unique frequency coefficients in the hypergraph spectrum domain, $\tilde{\mathcal{X}}_{F_s}$. We compare our results with the recently proposed CP-based HGSP framework [9]. Given that finding the rank R is an NP-hard problem and the signal is not necessarily band-limited, $R = N$ was used in this experiment for the symmetric orthogonal CP approximation [35]. Fig. 10(a) illustrates the significant performance gains attained by the proposed t-HGSP framework over competing state-of-the-art methods.

As in [9], we also test the efficiency of t-HGSP sparse signal representation over the MovieLens data sets [58] where each movie data point has rating scores and tags from viewers. For this numerical experiment, the scores of movies are considered as signals and the graph model is constructed based on the tag relationships, i.e., two or more movies are connected in a hyperedge if they have similar tags [9]. For convenience, we used a subset of 400 movies and we set $M = 3$. With the graph and hypergraph models, the sparse hypergraph signal representation is obtained using the same method explained above. Results depicted in Fig. 10(b) demonstrate that the proposed t-HGSP framework outperforms competing state-of-the-art methods.

Additionally, we consider a hyper-spectral point cloud sensed by NASA’s Goddard’s LiDAR, Hyperspectral and Thermal (G-LiHT) airborne imaging system [26], [59]. A hypergraph with a 3rd-order adjacency tensor \mathcal{A} was then built based on the similarities of the elements in the point cloud ($N = 882$). The *hypergraph signal* $\tilde{\mathcal{X}}_s$ was built from the set of 1-D signals in the hyperspectral point cloud according to Definition 8. The sparsified signal was obtained as before. To compare our results to the CP hypergraph-based approach [9], [35] and GSP [45], a sparse signal representation for each 1-D signal is obtained independently. The simple graph [45] is also built based on similarities. Fig. 10(c) illustrates large performance gains attained by the t-HGSP framework over state-of-the-art methods which demonstrates the benefit of jointly processing multiple signals.

B. Spectral Clustering

Clustering is key in many applications such as social network analysis, computer vision, and computational biology [60]. In

Algorithm 2: HGSP Fourier Spectral Clustering.

Input: Hypergraph \mathbf{H} and number of clusters K .

Output: Partitions of the hypergraph dataset S_1, \dots, S_K .

- 1: Compute the symmetrized Laplacian tensor \mathcal{L}_s .
- 2: Compute the first K eigen-matrices
 $\mathcal{V}_{[K]} = [\tilde{\mathcal{V}}_1, \tilde{\mathcal{V}}_2, \dots, \tilde{\mathcal{V}}_K]$ of \mathcal{L}_s .
- 3: Let $\hat{\mathbf{V}}_{[K]}^{(1)} = \hat{\mathcal{V}}_{[K]}(:, :, 1) \in \mathbb{R}^{N \times K}$ be the first frontal slice of the tensor $\hat{\mathcal{V}}_{[K]}$ that contains the K eigen-matrices in the Fourier domain.
- 4: For $i = 1, \dots, N$, let $\mathbf{y}_i \in \mathbb{R}^{1 \times K}$ be the i -th row of $\hat{\mathbf{V}}_{[K]}^{(1)}$.
- 5: Cluster the N nodes with features vectors $(\mathbf{y}_i)_{i=1, \dots, N}$ in \mathbb{R}^K with the k-means algorithm into K clusters $\{C_1, \dots, C_K\}$.
- 6: Use the clustering result to partition the hypergraph into S_1, \dots, S_K such as $S_i = \{\mathbf{v}_j | \mathbf{y}_j \in C_i\}$, i.e. the j -th node belongs to the i -th partition if $\mathbf{y}_j \in C_i$.

recent years, spectral clustering has aroused more and more attention due to its good clustering performance and solid theoretical foundation [61]. However, standard spectral clustering methods consider only pairwise connections between nodes and there are many applications in which interactions involving more than two nodes occur. Hence, it is fundamental to extend spectral clustering to hypergraphs. The definition of a suitable spectral space is key when developing a hypergraph spectral clustering. Li et al. [62] and Ahn et al. [60] proposed a hypergraph spectral clustering approach based on the eigenspace of what they called the hypergraph processed similarity matrix. Zhang et al. [9] introduced a hypergraph spectral clustering approach based on the hypergraph Fourier space given by the symmetric orthogonal CP decomposition. In contrast, we generalize graph spectral clustering techniques to our t-HGSP framework by means of the hypergraph Fourier space of the Laplacian tensor (Algorithm 2). As in [9], we used the zoo dataset [63] to create an $M = 3$ hypergraph and compare our results with those from the hypergraph similarity method (HSC) in [60] and CP-based HGSP [9]. The average Silhouette [64] of nodes was used to asses clustering quality among the different methods. This metric is particularly useful if the ground truth labels are not known. The silhouette value for each point is a measure of how similar that point is to points in its own cluster, when compared to points in other clusters. The silhouette value ranges from -1 to 1. A high average silhouette indicates a good clustering. Given that the ground truth labels are known, we also used the Normalized Mutual Information (NMI), weighted F1 score, and Jaccard index to compare the clustering performance [65]. In addition to the already mentioned advantages of our framework, in Figs. 10(d) and 12(a), it is shown that the proposed hypergraph Fourier space of the Laplacian tensor is not only suitable for hypergraph spectral clustering but also outperforms the prior art.

Additionally, to demonstrate the performance of the proposed framework on higher-order ($M > 3$) hypergraphs, we consider different subsets of two real-world dataset, the coauthorship networks, cora and DBLP, as depicted in Fig. 11. In this hypergraph, the nodes represent documents and all documents co-authored

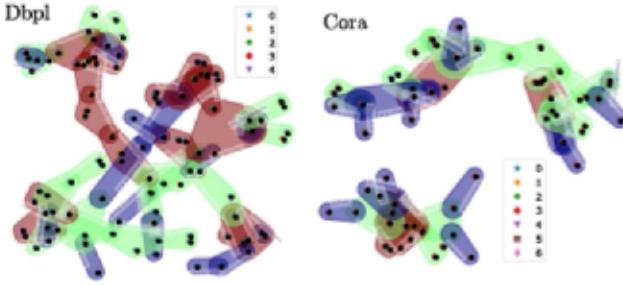


Fig. 11. Hypergraphs on a subset of two real-world co-authorship networks (left) DBLP ($M = 4, N = 59$) and (right) Cora ($M = 4, N = 96$). Nodes represent documents and all documents co-authored by an author are in one hyperedge [66]. The labels classify nodes (papers) into categories such as “algorithms” and “data mining”. Hyperedges’ cardinality is color-coded: (red) $|e| = 4$, (green) $|e| = 3$, and (blue) $|e| = 2$.

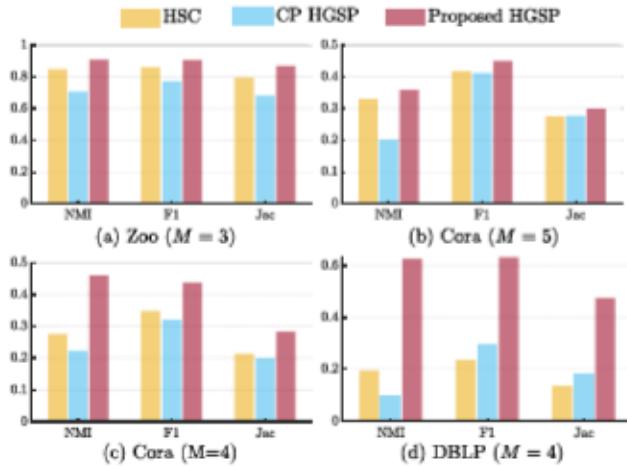


Fig. 12. Performance comparison of hypergraph spectral clustering on (a) the Zoo dataset ($M = 3, N = 101$), and different subsets of co-authorship networks: (b) Cora ($M = 5, N = 30$), (c) Cora ($M = 4, N = 59$), and (d) DBLP ($M = 4, N = 96$).

by an author are in one hyperedge [66]. The labels classify nodes (papers) into categories such as “algorithms”, “data mining”, and so on. For the higher-order orthogonal symmetric CP decomposition, we used the algorithm proposed in [35]. As observed in Fig. 12(b)-(d), the proposed approach outperforms HSC [60] and CP HGSP [9] when high-order hypergraphs are considered. More details about the spectral clustering algorithm, metrics, and other results can be found in the supplemental material.

C. Denoising

Hypergraph filters can be used to denoise a hypergraph signal. Let $z = x + n \in \mathbb{R}^N$ be a one dimensional noisy signal with noise n and $\tilde{z}_s \in \mathbb{R}^{N \times 1 \times N_s}$ its corresponding hypergraph signal. Assuming that the original signal is smooth on the graph, a simple approach to denoising is by spectrum shrinkage where filtering reduces to hard-thresholding of the spectral coefficients [67] such that

$$q_{F_s}(\lambda_l)^{(1)} = \begin{cases} 1, & \text{if } \lambda_l \leq \lambda_c \\ 0, & \text{otherwise,} \end{cases} \quad (37)$$

TABLE I
VISUAL QUALITY OF DENOISED IMAGES

Metric	GSP	CP HGSP	Proposed t-HGSP
Gaussian Distribution: $N(0,0.01)$			
PSNR	24.5687	21.0531	25.2840
MSE	0.0035	0.0078	0.0030
MSSIM	0.9361	0.8918	0.9414



Fig. 13. (a) Zoom of the original Cameraman image. The image was contaminated with (b) Gaussian noise and denoised with (c) GSP, (d) CP HGSP and (e) the proposed t-HGSP framework.

where λ_c is a cut-off frequency. Then, the denoised signal is $\tilde{x}_s = Q_s * \tilde{z}_s$. Since the frequency response of a clean signal is expected to be dominated by low-frequency components, the high-frequency components are most likely to be generated by noise. This noise is thus attenuated by hard-thresholding the transform coefficients [67]. To test, the performance of our approach, we conducted a denoising experiment on the “Cameraman” image. The experimental setup is similar to the one in the compression experiment, except for noise addition and the use of overlapping windows. The same hard-thresholding filter design was used for GSP and CP-based HGSP denoising. We contaminated the signal with Gaussian noise. The cut-off frequency was optimized for the best performance in all the cases. The proposed hypergraph framework outperforms CP-based HGSP and GSP denoising as shown in Table I. Even though almost no improvement over GSP is observed on the performance metrics, as shown in Fig. 13, the details of the image are better recovered by the proposed approach. A bigger section of the image is included in Fig. 14 of the Supplemental material.

IX. CONCLUSION

A novel tensor-based t-HGSP framework using t-product decompositions is introduced. The proposed framework not only generalizes traditional GSP to high-dimensional hypergraphs but more importantly is loss-free, preserves the dimensionality of the signals, and is less computationally complex when compared to the state-of-the-art. The core elements of the new t-HGSP framework have been laid down, including the concept of a hypergraph signal, hypergraph shifting, frequency analysis, and band-limited signals. Additionally, fundamental tools such as sampling and filtering were introduced and demonstrated experimentally. Notably, the proposed framework outperforms state-of-the-art by considering high-order properties of one-dimensional signals or jointly processing a set of signals, as demonstrated in the denoising and sparse signal representation experiments. Additionally, natural hypergraphs such as co-authorship networks can be exploited for applications

such as classification and clustering without being a map to a low-dimensional graph. However, the key to the success of hypergraph-based methods is having a meaningful hypergraph that may only be readily available for some applications. Hence, our current work is focused on learning optimal hypergraph topologies from data that could further boost the performance of the proposed t-HGSP tools.

Given the proposed t-HGSP framework, many opportunities emerge to continue exploring HGSP and its applications. For instance, further interpretations on the t-HGSP spectrum could be drawn by considering its connection with manifolds and the connection of GSP with other areas such as information theory. Moreover, other tensor decompositions such as HOSVD [68], tensor-train [69], and O-SVD [70] could be potentially used to define other HGSP frameworks with different properties. Our future work is focused on optimizing sampling patterns for different sampling operator configurations, hypergraph neural networks, scalability, and generalizing hypergraph's shifting operators to tensors that are not symmetric under the t-product algebra.

APPENDIX A PROOF OF THEOREM 1

Let $\mathcal{F}_s = \mathcal{A}_s$ and λ_i and λ_j be two eigen-tuples of \mathcal{F}_s with corresponding eigen matrices $\vec{\mathcal{V}}_i$ and $\vec{\mathcal{V}}_j$. Given that \mathcal{F}_s is symmetric, it has real eigen-tuples. If the eigen-tuples are ordered as $\lambda_i > \lambda_j$, then the total variations of their eigenvectors satisfy $\text{TV}_{\mathcal{H}}(\vec{\mathcal{V}}_i) < \text{TV}_{\mathcal{H}}(\vec{\mathcal{V}}_j)$.

Proof: Let $\vec{\mathcal{V}}_r$ be any eigenvector of \mathcal{F}_s , then the total variation is calculated as

$$\begin{aligned} \text{TV}_{\mathcal{H}}(\vec{\mathcal{V}}_r) &= \|\vec{\mathcal{V}}_r - \mathcal{F}_s^{\text{norm}} * \vec{\mathcal{V}}_r\| \\ &= \|\vec{\mathcal{V}}_r - \vec{\mathcal{V}}_r * \lambda_r * \text{abs}(\lambda_{\max})^{-1}\| \\ &= \|\vec{\mathcal{V}}_r\| * \text{abs}(\mathbf{e}_1 - \lambda_r * \text{abs}(\lambda_{\max})^{-1}) \\ &= \text{abs}(\mathbf{e}_1 - \lambda_r * \text{abs}(\lambda_{\max})^{-1}) \end{aligned} \quad (38)$$

Then, $\text{TV}_{\mathcal{H}}(\vec{\mathcal{V}}_j) - \text{TV}_{\mathcal{H}}(\vec{\mathcal{V}}_i) = (\lambda_i - \lambda_j) * \text{abs}(\lambda_{\max})^{-1}$. Thus, $\text{TV}_{\mathcal{H}}(\vec{\mathcal{V}}_j) > \text{TV}_{\mathcal{H}}(\vec{\mathcal{V}}_i)$ iff $\lambda_j < \lambda_i$.

APPENDIX B PROOF OF LEMMA 1

If $\vec{\mathcal{X}}_s$ is a K-band-limited signal then

$$\vec{\mathcal{X}}_s = \mathcal{V}_{[K]} * (\vec{\mathcal{X}}_{F_s})_{[K]}, \quad (39)$$

where $\mathcal{V}_{[K]} = [\vec{\mathcal{V}}_1, \dots, \vec{\mathcal{V}}_K] \in \mathbb{R}^{N \times K \times 2N+1}$ and $(\vec{\mathcal{X}}_{F_s})_{[K]} \in \mathbb{R}^{K \times 1 \times 2N+1}$ correspond to the K nonzero elements of the hypergraph signal in the frequency domain $\vec{\mathcal{X}}_{F_s}$.

Proof: Since $\vec{\mathcal{X}}_s$ is band-limited, then $(\mathbf{x}_{F_s})_{l,1} = \vec{\mathcal{V}}_l * \vec{\mathcal{X}}_s = \mathbf{0}$ when $l > K$. Then, we have that

$$\vec{\mathcal{X}}_s = \mathcal{V} * \vec{\mathcal{V}} * \vec{\mathcal{X}}_s \quad (40)$$

$$= \sum_{l=1}^K \vec{\mathcal{V}}_l * \underbrace{\vec{\mathcal{V}}_l^T * \vec{\mathcal{X}}_s}_{(\mathbf{x}_{F_s})_{l,1}} + \sum_{l=K+1}^N \vec{\mathcal{V}}_l * \underbrace{\vec{\mathcal{V}}_l^T * \vec{\mathcal{X}}_s}_{\mathbf{0}} \quad (41)$$

$$= \mathcal{V}_{[K]} * (\vec{\mathcal{X}}_{F_s})_{[K]} \quad (42)$$

APPENDIX C PROOF OF THEOREM 2

Let $\mathcal{V}_{[K]}$ be the tensor formed by K lateral slices of \mathcal{V} , i.e. the orthonormal set $\{\vec{\mathcal{V}}_1, \vec{\mathcal{V}}_2, \dots, \vec{\mathcal{V}}_K\}$, and let the sampling operator $\Psi \in \mathbb{R}^{Q \times N \times N_s}$ satisfy

$$\rho(\Psi * \mathcal{V}_{[K]})^{(k)} = K, \quad (43)$$

for $i = 1, 2, \dots, 2N + 1$. Since $\rho(\Psi * \mathcal{V}_{[K]})^{(k)} \leq \min(Q, K)$, the sample size Q has to be greater than or equal than the bandwidth K . The interpolation operator $\Phi = \mathcal{V}_{[K]} * \mathcal{U} \in \mathbb{R}^{N \times Q \times N_s}$ with $\mathcal{U} * \Psi * \mathcal{V}_{[K]} = \mathcal{I}_{K \times K \times N_s}$, achieves perfect recovery, i.e. $\vec{\mathcal{X}}_s = \Phi * \Psi * \vec{\mathcal{X}}_s$ for any K -bandlimited signal.

Proof: To proof the theorem, we show that $\mathcal{P} = \Phi * \Psi$ is a projector operator and that the range of Φ is the space of K -bandlimited signals as in [44]. Since $\rho(\Psi * \mathcal{V}_{[K]})^{(k)} = K$ for $i = 1, 2, \dots, 2N + 1$ and $\rho(\mathcal{U} * \Psi * \mathcal{V}_{[K]})^{(k)} = K$ for $i = 1, 2, \dots, 2N + 1$, then $\rho(\mathcal{U})^{(k)} = K$ for $i = 1, 2, \dots, 2N + 1$. Thus, the interpolation operator $\Phi = \mathcal{V}_{[K]} * \mathcal{U} \in \mathbb{R}^{N \times M \times N_s}$ spans the set of K -bandlimited signals. To prove that \mathcal{P} is a projector, we show that $\mathcal{P}^2 = \mathcal{P} * \mathcal{P} = \mathcal{P}$ as

$$\mathcal{P} * \mathcal{P} = \Phi * \Psi * \Phi * \Psi, \quad (44)$$

$$= \mathcal{V}_{[K]} * \mathcal{U} * \Psi * \mathcal{V}_{[K]} * \mathcal{U} * \Psi, \quad (45)$$

$$\stackrel{(a)}{=} \mathcal{V}_{[K]} * \mathcal{U} * \Psi = \mathcal{P}, \quad (46)$$

where (a) follows from $\mathcal{U} * \Psi * \mathcal{V}_{[K]} = \mathcal{I}_{K \times K \times N_s}$. Notice that when $M = K$, \mathcal{U} is the inverse of $\Psi * \mathcal{V}_{[K]}$; and when $M > K$, \mathcal{U} is the pseudo-inverse of $\Psi * \mathcal{V}_{[K]}$ as in the traditional matrix-based sampling theory.

REFERENCES

- [1] S. Klamt, U.-U. Haus, and F. Theis, "Hypergraphs and cellular networks," *PLoS Comput. Biol.*, vol. 5, no. 5, 2009, Art. no. e1000385.
- [2] O. Sporns, "Graph theory methods: Applications in brain networks," *Dialogues Clin. Neurosci.*, vol. 20, no. 2, pp. 111–121, 2018.
- [3] S. Derrible and C. Kennedy, "Applications of graph theory and network science to transit network design," *Transport Rev.*, vol. 31, no. 4, pp. 495–519, 2011.
- [4] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca, "Network analysis in the social sciences," *Science*, vol. 323, no. 5916, pp. 892–895, 2009.
- [5] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
- [6] 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.
- [7] A. Ortega, *Introduction to Graph Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2021.
- [8] M. M. Wolf, A. M. Klinvex, and D. M. Dunlavy, "Advantages to modeling relational data using hypergraphs versus graphs," in *Proc. IEEE High Perform. Extreme Comput. Conf.*, 2016, pp. 1–7.

[9] S. Zhang, Z. Ding, and S. Cui, "Introducing hypergraph signal processing: Theoretical foundation and practical applications," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 639–660, Jan. 2020.

[10] S. Zhang, S. Cui, and Z. Ding, "Hypergraph-based image processing," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 216–220.

[11] S. Barbarossa and M. Tsitsvero, "An introduction to hypergraph signal processing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2016, pp. 6425–6429.

[12] M. T. Schaub, Y. Zhu, J.-B. Seby, T. M. Roddenberry, and S. Segarra, "Signal processing on higher-order networks: Livin' on the edge... and beyond," *Signal Process.*, vol. 187, 2021, Art. no. 108149.

[13] C. Giusti, R. Ghrist, and D. S. Bassett, "Two's company, three (or more) is a simplex," *J. Comput. Neurosci.*, vol. 41, no. 1, pp. 1–14, 2016.

[14] B. Heintz and A. Chandra, "Enabling scalable social group analytics via hypergraph analysis systems," in *Proc. 7th USENIX Workshop Hot Topics Cloud Comput. (HotCloud 15)*, 2015, pp. 14–14.

[15] S. Zhang, S. Cui, and Z. Ding, "Hypergraph spectral analysis and processing in 3D point cloud," *IEEE Trans. Image Process.*, vol. 30, pp. 1193–1206, 2021.

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

[17] S. Zhang, Q. Deng, and Z. Ding, "Introducing graph signal processing over multilayer networks: Theoretical foundations and frequency analysis," 2021, *arXiv:2108.13638*.

[18] N. Shahid, F. Grassi, and P. Vandergheynst, "Tensor robust PCA on graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 5406–5410.

[19] T. Michoel and B. Nachtergaelle, "Alignment and integration of complex networks by hypergraph-based spectral clustering," *Phys. Rev. E*, vol. 86, no. 5, 2012, Art. no. 056111.

[20] Z. Wang and M. Menenti, "Challenges and opportunities in lidar remote sensing," *Front. Remote Sens.*, vol. 2, 2021, Art. no. 641723. [Online]. Available: <https://www.frontiersin.org/article/10.3389/frsen.2021.641723>

[21] Y. Yu, M. M. d'Alessandro, S. Tebaldini, and M. Liao, "Signal processing options for high resolution SAR tomography of natural scenarios," *Remote Sens.*, vol. 12, no. 10, 2020, Art. no. 1638. [Online]. Available: <https://www.mdpi.com/2072-4292/12/10/1638>

[22] H. Rueda, C. Fu, D. L. Lau, and G. R. Arce, "Single aperture spectral ToF compressive camera: Toward hyperspectral depth imagery," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 7, pp. 992–1003, Oct. 2017.

[23] H. Rueda-Chacon, J. F. Florez-Ospina, D. L. Lau, and G. R. Arce, "Snapshot compressive ToF spectral imaging via optimized color-coded apertures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2346–2360, Oct. 2020.

[24] B. Ndagano et al., "Imaging and certifying high-dimensional entanglement with a single-photon avalanche diode camera," *NPJ Quantum Inf.*, vol. 6, no. 1, pp. 1–8, 2020.

[25] E. Mazarico, M. K. Barker, G. A. Neumann, M. T. Zuber, and D. E. Smith, "Detection of the lunar body tide by the lunar orbiter laser altimeter," *Geophys. Res. Lett.*, vol. 41, no. 7, pp. 2282–2288, 2014.

[26] B. D. Cook et al., "Nasa goddard's lidar, hyperspectral and thermal (g-LIHT) airborne imager," *Remote Sens.*, vol. 5, no. 8, pp. 4045–4066, 2013. [Online]. Available: <https://www.mdpi.com/2072-4292/5/8/4045>

[27] K. Fukunaga, S. Yamada, H. S. Stone, and T. Kasai, "A representation of hypergraphs in the euclidean space," *IEEE Comput. Archit. Lett.*, vol. 33, no. 04, pp. 364–367, Apr. 1984.

[28] A. Banerjee, A. Char, and B. Mondal, "Spectra of general hypergraphs," *Linear Algebra Appl.*, vol. 518, pp. 14–30, 2017.

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

[30] T. Biyikoglu, J. Leydold, and P. Stadler, *Laplacian eigenvectors of graphs: Perron-frobenius and faber-krahn type theorems* (Lecture Notes in Mathematics). Berlin, Heidelberg: Springer, 2007. [Online]. Available: <https://books.google.com/books?id=LnUbExxLviOC>

[31] A. Brouwer and W. Haemers, *Spectra of graphs* (Universitext). New York, NY, USA: Springer, 2011. [Online]. Available: <https://books.google.com/books?id=P98THwYgrXYC>

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

[33] A. Hatcher, *Algebraic Topology*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[34] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*. vol. 45, Amsterdam, The Netherlands: Elsevier, 1984.

[35] J. Pan and M. K. Ng, "Symmetric orthogonal approximation to symmetric tensors with applications to image reconstruction," *Numer. Linear Algebra Appl.*, vol. 25, no. 5, 2018, Art. no. e2180.

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

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

[38] C. J. Hillar and L.-H. Lim, "Most tensor problems are np-hard," *J. ACM (JACM)*, vol. 60, no. 6, pp. 1–39, 2013.

[39] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.

[40] C. D. Martin, R. Shafer, and B. LaRue, "An order-p tensor factorization with applications in imaging," *SIAM J. Sci. Comput.*, vol. 35, no. 1, pp. A474–A490, 2013.

[41] A. Bretto and L. Gillibert, "Hypergraph-based image representation," in *Proc. 5th Int. Workshop Graph-Based Representations Pattern Recognit.*, 2005, pp. 1–11.

[42] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013.

[43] D. F. Gleich, C. Greif, and J. M. Varah, "The power and arnoldi methods in an algebra of circulants," *Numer. Linear Algebra Appl.*, vol. 20, no. 5, pp. 809–831, 2013.

[44] 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, Jun. 2014.

[45] 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.

[46] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3584–3598, Jul. 2018.

[47] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, Jul. 2016.

[48] D. L. Lau, G. R. Arce, A. Parada-Mayorga, D. Dapena, and K. Pena-Pena, "Blue-noise sampling of graph and multigraph signals: Dithering on non-euclidean domains," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 31–42, Nov. 2020.

[49] A. Parada-Mayorga, D. L. Lau, J. H. Giraldo, and G. R. Arce, "Sampling of graph signals with blue noise dithering," in *Proc. IEEE Data Sci. Workshop*, 2019, pp. 150–154.

[50] A. Parada-Mayorga, D. L. Lau, J. H. Giraldo, and G. R. Arce, "Blue-noise sampling on graphs," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 5, no. 3, pp. 554–569, Sep. 2019.

[51] D. Dapena, D. L. Lau, and G. R. Arce, "Density aware blue-noise sampling on graphs," in *Proc. IEEE 30th Eur. Signal Process. Conf.*, 2022, pp. 1766–1770.

[52] D. Dapena, K. Pena-Pena, D. L. Lau, and G. R. Arce, "Generalized blue-noise sampling on graphs," *IEEE J. Eng. Med. Biol.*, submitted.

[53] D. Dapena, D. L. Lau, and G. R. Arce, "Parallel graph signal processing: Sampling and reconstruction," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 9, pp. 190–206, 2023.

[54] J. F. Florez-Ospina, A. K. Alrushud, D. L. Lau, and G. R. Arce, "Block-based spectral image reconstruction for compressive spectral imaging using smoothness on graphs," *Opt. Exp.*, vol. 30, no. 5, pp. 7187–7209, 2022.

[55] J. F. Florez-Ospina, D. L. Lau, D. Guillot, K. Barner, and G. R. Arce, "Smoothness on rank-order path graphs and its use in compressive spectral imaging with side information," *Signal Process.*, vol. 201, 2022, Art. no. 108707.

[56] M. P. Sampat et al., "Computer-aided detection and diagnosis in mammography," *Handbook Image Video Process.*, vol. 2, no. 1, pp. 1195–1217, 2005.

[57] S. Sherman and T. G. Kolda, "Estimating higher-order moments using symmetric tensor decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 41, no. 3, pp. 1369–1387, 2020.

[58] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2015.

- [59] B. D. Cook et al., "NASA Goddard's LiDAR, hyperspectral and thermal (G-LiHT) airborne imager," *Remote Sens.*, vol. 5, no. 8, pp. 4045–4066, 2013.
- [60] K. Ahn, K. Lee, and C. Suh, "Hypergraph spectral clustering in the weighted stochastic block model," *IEEE J. Sel Top Signal Process.*, vol. 12, no. 5, pp. 959–974, Oct. 2018.
- [61] M. C. Nascimento and A. C. De Carvalho, "Spectral methods for graph clustering—a survey," *Eur. J. Oper. Res.*, vol. 211, no. 2, pp. 221–231, 2011.
- [62] 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.
- [63] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [64] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.
- [65] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proc. SIAM Int. Conf. Data Min. SIAM*, 2012, pp. 106–117.
- [66] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "Hypergcn: A new method for training graph convolutional networks on hypergraphs," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates Inc., 2019.
- [67] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *Proc. IEEE 15th Int. Workshop Multimed. Signal Process.*, 2013, pp. 001–006.
- [68] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [69] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [70] C. Zeng and M. K. Ng, "Decompositions of third-order tensors: Hosvd, t-SVD, and beyond," *Numer. Linear Algebra Appl.*, vol. 27, no. 3, 2020, Art. no. e2290.



Karelia Pena-Pena (Member, IEEE) received the B.Sc. degree in electrical engineering from Universidad de Los Andes, Merida, Venezuela, in 2017, the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Delaware, Newark, DE, USA, in 2020 and 2023, respectively. She is currently a Senior Data Scientist with Intuit, Mountain View, CA, USA. Her research interests include graph signal processing, natural language processing, computer vision, machine learning, and optimization.



Daniel L. Lau (Senior Member, IEEE) received the B.Sc. degree (with highest distinction) in electrical engineering from Purdue University, West Lafayette, IN, USA, and the Ph.D. degree from the University of Delaware, Newark, DE, USA, in 1995 and 1999, respectively. He is currently a Professor and the Director of graduate studies for electrical and computer engineering with the University of Kentucky, Lexington, KY, USA. His research interests include image and signal processing, 3D imaging, and machine vision. His work has also been featured in trade magazines such as *Imaging Insight*, *Prosilica Camera News*, and *Inspect Magazine*.



Gonzalo R. Arce (Life Fellow, IEEE) is currently the Charles Black Evans Distinguished Professor of electrical and computer engineering, and J.P. Morgan-Chase Senior Faculty Fellow with the University of Delaware, Newark, DE, USA. He holds 25 U.S. patents and is the co-author of four books. His research interests include computational imaging, data science, and machine learning. He held the 2010 and 2017 Fulbright-Nokia Distinguished Chair of Information and Communications Technologies with Aalto University, Espoo, Finland. He was the recipient of the NSF Research Initiation Award. He was elected Fellow of the IEEE, SPIE, and the National Academy of Inventors.