Hypergraph Similarity Measures

Amit Surana, *Member, IEEE*, Can Chen, and Indika Rajapakse, *Member, IEEE*

***Abstract*—In this paper we present a novel framework for hypergraph similarity measures (HSMs) for hypergraph comparison. Hypergraphs are generalizations of graphs in which edges may connect any number of vertices, thereby representing multi-way relationships which are ubiquitous in many real-world networks including neuroscience, social networks, and bioinformatics. We propose two approaches for developing HSMs. The first approach is based on transforming the hypergraph into a graph representation, e.g., clique and star expansion, and then invoking the standard graph similarity measures. The second approach relies on a tensor-based representation of hypergraphs which intrinsically captures multi-way relations, and define similarity measures using tensor algebraic notions. Within each approach we present a collection of measures which either assess hypergraph similarity at a specific scale e.g., local, mesoscopic or global, or provide a more holistic multi-scale comparison. We discuss the advantages and disadvantages of the two proposed approaches, and demonstrate their performance on synthetic hypergraphs and hypergraphs derived from experimental biological datasets.**

***Index Terms*—Hypergraphs, similarity measures, tensors, biological systems.**

I. Introduction

COMPLEX systems in sociology, biology, cyber-security, telecommunications, and physical infrastructure are often represented as a set of entities, i.e., "vertices" with binary relationships or "edges," and hence are analyzed via graph theoretic methods. Graph models, while simple and to some degree universal, are limited to representing pairwise relationships between entities. However, real-world phenomena can be rich in multi-way relationships, dependencies between more than two variables, or properties of collections of more than two objects. Examples include computer networks where the dynamic relations are defined by packets exchanged over time between computers, co-authorship networks where relations are articles written by two or more authors, historical documents where multiple persons can be mentioned together, brain activity where multiple regions can be highly active at the same time, film actor networks, and protein-protein interaction networks [1], [2], [3], [4]. Furthermore, hypergraph representation and learning has recently attracted increasing attention due to its flexibility and capability in modeling complex data correlation, and is being utilized in a diverse set of applications, including computer vision [5], medical imaging [6], recommendation systems [7], and mode seeking in graphs [8], [9], see [10] for a recent review.

A hypergraph is a generalization of a graph in which its hyperedges can join any number of vertices [11]. Thus, hypergraphs can capture multi-way relationships unambiguously [12], and are a natural representation of a broad range of systems mentioned above. Although an expanding body of research attests to the increased utility of hypergraph-based analyses, many network science methods have been historically developed explicitly (and often, exclusively) for graph-based analyses and do not directly translate to hypergraphs. Consequently, new frameworks are being developed for representation, learning, and analysis of hypergraphs, see [2], [3] for a recent survey. These include techniques for converting hypergraphs into graphs and defining hypergraph Laplacian [13], higher-order random walks-based hypergraph analysis [14], and defining dynamics on hypergraphs [15].

As tensors [16] provide a natural framework to represent multi-dimensional patterns and capture higher-order interactions, they are finding increasing role in context of hypergraphs. For example, the spectral theory of graphs has been extended to hypergraphs using tensor eigenvalues [17], and authors in [18] define notion of tensor entropy for uniform hypergraphs generalizing von Neumann entropy of a graph to hypergraphs. The problem of controllability of dynamics on hypergraphs is studied via tensor-based representation and nonlinear control theory in [19]. Similar to above mentioned work, the goal of this paper is to extend the graph comparison framework to hypergraphs.

Comparison of structures such as modular communities, hubs, and trees yield insight into the generative mechanisms and functional properties of the graph. Graph comparison can be used for comparing brain or metabolic networks for different subjects, or the same subject before and after a treatment, and for characterizing the temporal network evolution during treatment [20]. Classification of graphs, for example in context of protein-protein interaction networks and online social networks can be facilitated via use of graph comparison measures [21]. Combined with a clustering algorithm, a graph comparison measure can be used to aggregate networks in a meaningful way and reveal redundancy in the data/networks [22]. Graph comparison can be

Manuscript received 13 April 2022; revised 10 September 2022; accepted 19 October 2022. Date of publication 27 October 2022; date of current version 23 February 2023. This work was supported by the Air Force Office of Scientific Research under Award FA9550-18-1-0028. Recommended for acceptance by Dr. Xuelong Li. *(Corresponding author: Amit Surana.)*

Amit Surana is with the Raytheon Technologies Research Center, East Hartford, CT 06108 USA (e-mail: amit.surana@rtx.com).

Can Chen is with the Department of Mathematics, University of Michigan, Ann Arbor, MI 48109 USA, and also with the Channing Division of Network Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, MA 02115 USA (e-mail: canc@umich.edu).

Indika Rajapakse is with the Department of Computational Medicine & Bioinformatics, Medical School and the Department of Mathematics, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: indikar@umich.edu).

Digital Object Identifier 10.1109/TNSE.2022.3217185

used for evaluating the accuracy of statistical or generative network models [23], and can be further utilized as an objective function to drive the optimization procedure to fit graph models to data.

In order to compare graphs, a variety of similarity measures (SM) or distances have been proposed in the literature which either assess similarity at a specific scale e.g., local or global, or provide a more holistic multi-scale comparison. See references [20], [24], [25] for a comprehensive review. While there is a rich body of literature for graph similarity measures (GSM), analogous notions for hypergraph are lacking in literature. To address this gap, we propose two new approaches for comparison of undirected and weighted hypergraphs. Just like for GSMs within each of these hypergraph similarity measure (HSM) approaches we present a collection of SMs which either assess hypergraph similarity at a specific scale or provide a multi-scale comparison. Specifically, the key contributions of this paper are as follows:

- We develop an indirect approach for comparing hypergraphs by first transforming the hypergraph into a graph and then invoking standard GSMs. In particular we explore clique and star expansion for this transformation. While information about hypergraph structure may be lost during such transformations/projections, the assumption is that relevant salient features may still be preserved which are sufficient to capture key differences between underlying hypergraphs. We refer to these SMs as indirect HSMs.
- We introduce another direct approach which relies on tensor-based representation of hypergraphs which intrinsically captures the multi-way relations encoded by its hyperedges. In particular we use adjacency tensor and Laplacian tensor associated with hypergraphs, and tensor algebraic notions of tensor eigenvalues/eigenvectors and higher-order singular values to develop new notions of SMs for hypergraphs. We refer to these SMs as direct HSMs.
- We test the proposed HSMs on synthetic hypergraphs to assess their usability in discerning between common hypergraph topologies. We also apply the methods to experimental hypergraph datasets arising in biological applications.

The paper is organized into seven sections. We introduce basic notation and mathematical preliminaries related to hypergraphs, and discuss some desirable characteristics of HSMs in Section II. In Section III, we provide a short survey of different GSMs. We then use these GSMs in Section IV to define indirect HSMs based on conversion of hypergraphs into graphs. In Section V, we develop notions of direct HSMs using tensor-based representation of hypergraphs. Applications to synthetic and experimental hypergraph datasets are presented in Section VI. We discuss pros/cons of indirect and direct HSMs and directions for future research in Section VI-D, and conclude in Section VII.

II. Preliminaries

A. Hypergraph

Let V be a finite set. A *undirected hypergraph* $\mathcal{G}$ is a pair (V, E) where $E \subseteq \mathcal{P}(V)\backslash\{\emptyset\}$, the power set of V. The elements of V are called the vertices, and the elements of E are called the hyperedges. We note that in this definition of hypergraph we do not allow for repeated vertices within an hyperedge (often called hyperloops). For an *undirected weighted hypergraph* $(V, E, w(\cdot))$, there is positive weight function $w : E \to (0, \infty)$ which defines a weight $w(e) > 0$ associated with each hyperedge $e \in E$. The degree $d(v)$ of a vertex $v \in V$ is $d(v) = \sum_{e \in E | v \in e} w(e)$. The degree of an hyperedge e is denoted by $d(e) = |e|$, where $|\cdot|$ denotes set cardinality. For k-uniform hypergraphs, the degree of each hyperedge is the same, i.e. $d(e) = k$. The vertex-hyperedge incidence matrix $\mathbf{H}$ is a $|V| \times |E|$ matrix where the entry $h(v, e)$ is 1 if $v \in e$ and 0 otherwise. By these definitions, we have,

$$d(v) = \sum_{e \in E} w(e) h(v, e), \qquad d(e) = \sum_{v \in V} h(v, e).$$

Let $\mathbf{D}_e$ and $\mathbf{D}_v$ be the diagonal matrices consisting of hyperedge and vertex degrees as diagonal entries, respectively. Similarly we will denote by $\mathbf{W}$ the diagonal matrix formed by hyperedge weights $w(\cdot)$ as its diagonal entries. If not stated otherwise, we will assume that the hypergraph in consideration is undirected and weighted throughout this paper.

Note that a standard graph is a 2-uniform hypergraph. We will denote a standard *graph* by G, and by $\mathbf{A}$ as its adjacency matrix which is $|V| \times |V|$ matrix with entry (u, v) equal to the edge weight $w(e)$ (where e is such that $(u, v) \in e$) if they are connected, and 0 otherwise. The incidence matrix $\mathbf{H}$, and the diagonal matrices $\mathbf{D}_v$ and $\mathbf{W}$ are similar to as defined above.

B. Hypergraph Similarity Measure (HSM)

Let $\mathbb{G}$ be the space of hypergraphs with finite number of vertices. A *hypergraph similarity measure* (HSM) $\mathcal{D}$ is a symmetric non-negative function $\mathcal{D} : \mathbb{G} \times \mathbb{G} \to [0, \infty)$, i.e. $\mathcal{D}(\mathcal{G}, \tilde{\mathcal{G}}) = \mathcal{D}(\tilde{\mathcal{G}}, \mathcal{G})$ for any $\mathcal{G}, \tilde{\mathcal{G}} \in \mathbb{G}$. $\mathcal{D}$ quantifies distance between two hypergraphs with smaller values indicating higher degree of similarity.

Note that in general $\mathcal{D}$ is not required to satisfy $\mathcal{D}(\mathcal{G}, \tilde{\mathcal{G}}) = 0$ even when $\mathcal{G}$ and $\tilde{\mathcal{G}}$ are isomorphic, or the triangular inequality, and thus may not be a valid metric. But depending on the application, such requirements may be further imposed on $\mathcal{D}$. Approaches to graph comparison can be roughly divided into two groups, those that consider or require two graphs to be defined on the same set of vertices, and those that do not.

To distinguish SMs which have been defined specifically for graphs in the literature, we will refer to them as *graph similarity measures* (GSMs), and denote them by D.

C. Properties of Similarity Measures

In this section we summarize some desirable properties of graph similarity measures which have been noted in the literature, and can also be applied in context of hypergraphs. These

properties can serve as guidelines for selecting appropriate SM, and can further be modified and enriched by the data analyst depending on the application at hand. Examples of some desirable properties for the SMs include [26]:

- *Edge-importance* [*EI*]: modifications of the graph structure yielding disconnected components should be penalized more.
- *Edge-submodularity* [*ES*]: a specific change is more important in a graph with a few edges than in a denser graph on the same vertices.
- *Weight awareness* [*WA*]: the impact on the similarity measure increases with the weight of the modified edge.
- *Focus awareness* [*FA*]: random changes in graphs are less important than targeted changes of the same extent.

Depending on the application, additional invariance properties may be imposed on the SMs, such as [27]:

- *Permutation-invariance* [*PI*]: implies that if we permute the node indices the SM does not change. So if two graphs are isomorphic, then for a permutation invariant SM, the SM will be zero.
- *Scale-adaptivity* [*SA*]: implies that the SM accounts for differences in both local (edge and node) and global (community) features. Using local features only, a SM would deem two graphs sharing local patterns to have near-zero distance although their global properties (such a page-rank features) may differ, and, conversely, relying on global features only would miss the differences in local structure (such as connectivity of nodes locally to each other).
- *Size-invariance* [*SI*]: is the capacity of SM to discern if two graphs contain similar substructures irrespective of number of nodes in the graphs. For example, two social graphs with different number of nodes containing same fixed-size communities (e.g., criminal circles) should have a near-zero SM.

III. Review of Graph Similarity Measures

We review some key GSMs, the material is taken from the survey articles [24], [25], [28]. Approaches to graph comparison can be categorized from different perspectives.

One categorization is based on whether the graph comparison method requires the two graphs to be defined on the same set of vertices or not. The former eliminates the need to discover a mapping between node sets, making comparison relatively easier. A common approach for comparison without assuming node correspondence is to build the SM using graph invariants. Graph invariants are properties of a graph that hold for all isomorphs of the graph. Using an invariant mitigates any concerns with the encoding of the graphs, and the SM is instead focused completely on the graph topology.

Another categorization of graph comparison methods is based on scale at which they compare structures [28]. Local SMs are only sensitive to differences in direct neighbourhood of each node, while global SMs may ignore node identities and perceive differences only in global structures in the graph such as hubs, communities, number of spanning trees, etc. On the other hand, mesoscopic SMs work at intermediate scale such that they not only preserve vertex identities but also incorporate information characterising vertices by their relationship to the whole graph, rather than uniquely with respect to their neighbours. Finally, multi-scale SMs attempt to capture aspects from multiple scales i.e., local, global and/or mesoscopic in quantifying differences between graphs.

Let G and $\tilde{G}$ be two graphs under comparison with adjacency matrices $\mathbf{A}$ and $\tilde{\mathbf{A}}$, respectively, and let their graph Laplacians be $\mathbf{L}$ and $\tilde{\mathbf{L}}$, respectively. The graph Laplacian $\mathbf{L}$ (and similarly $\tilde{\mathbf{L}}$) could be the standard combinatorial Laplacian

$$\mathbf{L}_{un} = \mathbf{D}_v - \mathbf{A}, \tag{1}$$

or its normalized version,

$$\mathbf{L} = \mathbf{I} - \mathbf{D}_v^{-1/2}\mathbf{A}\mathbf{D}_v^{-1/2}. \tag{2}$$

We shall denote Laplacian eigenvalues as $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The literature remains divided on which version of the Laplacian to pick for defining the SM. Since the eigenvalues of the normalized Laplacian are bounded between 0 and 2, it makes it a more stable and preferable representation. Therefore, if otherwise stated, we will always use the normalized Laplacian $\mathbf{L}$ in definition of the GSMs.

- *Structural SMs* [28]: The simplest GSMs are obtained by directly computing the difference of the adjacency matrices of the two graphs and then using a suitable norm e.g., Euclidean, Manhattan, Canberra, or Jaccard. Examples of such GSMs include, the Hamming distance,

$$D_H(G, \tilde{G}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}|\mathbf{A}_{ij} - \tilde{\mathbf{A}}_{ij}|,$$

  and, the Jaccard distance,

$$D_J(G, \tilde{G}) = 1 - \frac{\sum_{ij}\min(\mathbf{A}_{ij}, \tilde{\mathbf{A}}_{ij})}{\sum_{ij}\max(\mathbf{A}_{ij}, \tilde{\mathbf{A}}_{ij})}.$$

  Structural SMs focus on differences in the direct local neighborhood of each node, and are agnostic to other more global structures in the graph.
- *Feature-based SMs* [29], [30]: Another possible method for comparing graphs is to look at specific "features" of the graph, such as the degree distribution, betweenness centrality distribution, diameter, number of triangles, number of k-cliques, etc. For graph features that are vector-valued (such as degree distribution) one might also consider the vector as an empirical distribution and take as graph features the sample moments (or quantiles, or other statistical properties). A feature-based distance is a distance that uses comparison of such features to compare graphs. If we are using node dependent features, the method aggregates a feature-vertex matrix of size $k \times n$, where k is number of features selected. This feature-vertex matrix for the two graphs can then be directly compared, or can be further reduced to a

"signature vector" that consists of the mean, median, standard deviation, skewness, and kurtosis of each feature across vertices. These signature vectors are then compared in order to obtain a SM between graphs. NETSIMILE [29] is an example of feature based distance which uses local and egonet-based features (e.g., degree, volume of egonet as fraction of maximum possible volume, etc.). In the neuroscience literature, in particular, feature-based methods are fairly popular.

In this paper, we will utilize node centrality vector $\mathbf{c} = (c_1, \ldots, c_n)^T$ as the feature for graph comparison, where superscript T denotes the vector/matrix transpose. Let $c_i, i = 1, \ldots, n$ and $\tilde{c}_i, i = 1, \ldots, n$ be normalized (i.e. $||\mathbf{c}||_1 = ||\tilde{\mathbf{c}}||_1 = 1$, where $||\cdot||_p$ is p-vector/matrix norm.) node centralities for the graphs G and $\tilde{G}$, respectively, then centrality-based SM is given by:

$$D_C(G, \tilde{G}) = \frac{1}{n} \sum_{i=1}^{n} |c_i - \tilde{c}_i|. \tag{3}$$

Note that one could use any notion of centrality, e.g., betweeness centrality, closeness centrality, eigenvector centrality, etc, as relevant for the application [30]. Since, centrality measures typically characterize vertices as either belonging to the core or to the periphery of the graph, and thus encode global topological information on the status of vertices within the graph, SMs based on centrality capture mesoscopic differences between graphs.

- *Spectral SMs* [28], [31]: Spectral SMs on the other hand are more suitable for analyses where the critical information in the graph structure is contained at a global scale, rather than locally. Spectral SMs are global measures defined using the eigenvalues of either the adjacency matrix $\mathbf{A}$ or of some version of the Laplacian $\mathbf{L}$. Both the eigenvalues of the Laplacian and those of the adjacency matrix can be related to physical properties of a graph, and can thus be considered as characteristics of its states. The adjacency matrix does not downweight any changes and treats all vertices equivalently. On the other hand, the eigenspectrum of the Laplacian accounts for the degree of the vertices and is known to be robust to most perturbations. Specific example of spectral SMs include l_p distance on space of Laplacian eigenvalues,

$$D_\lambda(G, \tilde{G}) = \frac{1}{n} \sum_{i=1}^{n} |\lambda_i - \tilde{\lambda}_i|^p, \tag{4}$$

and spanning tree SM,

$$D_{ST}(G, \tilde{G}) = |\log(T_G) - \log(T_{\tilde{G}})|, \tag{5}$$

where, T_G is number of spanning trees in the graph, given by

$$T_G = \frac{1}{n} \prod_{i=2}^{n} \lambda_i.$$

Other SMs include distances based on the eigenspectrum distributions,

$$D_\rho(G, \tilde{G}) = \int |\rho_G(x) - \rho_{\tilde{G}}(x)| dx, \tag{6}$$

where,

$$\rho_G(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\lambda_i)^2}{2\sigma^2}}.$$

Another related SM is the Ipsen–Mikhailov distance which characterizes the difference between two graphs by comparing their spectral densities, rather than the raw eigenvalues themselves.

- *DELTACON* [26]: This SM is based on the fast belief propagation method of measuring node affinities. It uses the fast belief propagation matrix

$$\mathbf{S} = [\mathbf{I} + \epsilon^2 \mathbf{D} - \epsilon \mathbf{A}]^{-1},$$

where, $0 < \epsilon \ll 1$, and compares the two representations $\mathbf{S}$ and $\tilde{\mathbf{S}}$ via the Matusita difference, leading to

$$\mathbf{D}_\Delta(G, \tilde{G}) = \frac{1}{n^2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\sqrt{S_{ij}} - \sqrt{\tilde{S}_{ij}} \right)^2 \right)^{1/2}. \tag{7}$$

Fast belief propagation is designed to model the diffusion of information throughout a graph, and so in theory should be able to capture differences in both global and local structures in the graph.

- *Heat Spectral Wavelets* [32]: An alternative is to derive characterizations of each node's topological properties through a signal processing approach. A specific example for this type of SM include the heat spectral wavelets in which the eigenvalues are modulated and combined with their respective eigenvectors to yield a "filtered" representation of the graph's signal. For a given scale factor $\tau > 0$, structural signature $\boldsymbol{\xi}_u$ for each node u is defined to be a vector of coefficients,

$$\boldsymbol{\xi}_u^\tau = (\Psi_{1,u}^\tau, \Psi_{2,u}^\tau, \ldots, \Psi_{n,u}^\tau)^T,$$

where,

$$\Psi_{v,u}^\tau = \sum_{i=1}^{n} e^{-\tau\lambda_i} V_{ui} V_{vi},$$

with, $\mathbf{L} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$ being the Laplacian's eigenvector decomposition. Let $\boldsymbol{\xi}_u = ((\boldsymbol{\xi}_u^{\tau_1})^T, \ldots, (\boldsymbol{\xi}_u^{\tau_m})^T)^T$ be the combined vector for a set of selected scales $\tau_1, \tau_2, \ldots, \tau_m$. By choosing these scales appropriately, one can capture information on the connectedness and centrality of each node within the network, thereby providing a way to encompass in a single Euclidean

vector all the necessary information to characterize vertices' topological status within the graph. Then the heat kernel SM between the graphs amounts to the average l_2 distance between corresponding node's structural embedding $\boldsymbol{\xi}_i$, i.e.,

$$D_{HK}(G,\tilde{G}) = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{\xi}_i - \tilde{\boldsymbol{\xi}}_i||_2. \tag{8}$$

- *NetLSD* [27]: Similar to heat spectral wavelet, in network Laplacian spectral descriptor (NetLSD) a heat kernel is defined as,

$$\mathbf{H}_\tau = e^{-\tau\mathbf{L}} = \mathbf{V}e^{-\boldsymbol{\Lambda}\tau}\mathbf{V}^T,$$

along with its heat trace,

$$h_\tau = \mathrm{Tr}(\mathbf{H}_\tau) = \sum_{j=1}^{n} e^{-\lambda_j\tau}.$$

Then the NetLSD condenses the graph representation in form of a heat trace signature $h(G) = \{h_\tau\}_{\tau>0}$, which comprises of a collection of heat traces at different time scales τ. The continuous-time function h_τ is finally transformed into a finite-dimensional vector by sampling τ over a suitable time interval. The SM between G and $\tilde{G}$ is then taken to be the l_∞ norm of vector difference between $h(G)$ and $h(\tilde{G})$, i.e.,

$$D_{LSD}(G,\tilde{G}) = ||h(G) - h(\tilde{G})||_\infty. \tag{9}$$

Note, that the heat kernel can be seen as continuous-time random walk propagation (where, $(\mathbf{H}_\tau)_{ij}$ is the heat transferred from node i to node j at time τ), and its diagonal (sometimes referred to as the autodiffusivity function or the heat kernel signature) can be seen as a continuous-time PageRank. As τ approaches zero, the Taylor expansion yields $\mathbf{H}_\tau = 1 - \mathbf{L}\tau$ meaning the heat kernel depicts local connectivity. On the other hand for large τ, $\mathbf{H}_\tau = 1 - e^{-\tau\lambda_2}\mathbf{v}_2\mathbf{v}_2^T$ where $\mathbf{v}_2$ is the Fielder vector, and it encodes global connectivity. Thus, the heat kernel localizes around its diagonal and the degree of localization (as captured by the heat trace) depends on the scale τ. It can thereby be tuned to capture both local and global graph structures.

- *Graph Embedding-based SMs* [33]: Given the diversity of structural features in graphs, and the difficulty of designing by hand the set of features that optimizes the graph embedding, several researchers have proposed recently to learn the embedding from massive datasets of existing networks. Such algorithms learn an embedding from a set of graphs into Euclidean space, and then compute a notion of similarity between the embedded representation of the graphs. All these approaches rely on the extension of convolutional neural networks to non Euclidean structures, such as manifolds and graphs. For example, the SimGNN [34] approach uses graph neural networks (GNNs) as a learnable embedding function that maps a graph into an Euclidean vector.
- *Graph Kernel-based SMs* [35]: A popular approach to learning with graph-structured data is to make use of graph kernels – functions which measure the similarity between graphs. These kernels can be used for comparing graphs. Many different graph kernels have been defined, which focus on different types of substructures in graphs, such as random walks, shortest paths, subtrees, and cycles. One particular approach is based on graphlets which are small, connected, non-isomorphic, induced subgraph of a larger graph. There are 30 graphlets with 2- to 5-vertices. Each graphlet contains "symmetrical vertices" which are said to belong to the same automorphism orbit. The automorphism orbits represent topologically different ways in which a graphlet can touch a node. The Graphlet Degree Vector (GDV) of a node generalises the notion of a node's degree into a 73-dimensional vector where each of the 73 components of that vector captures the number of times node n is touched by a graphlet at orbit i. Using GDV one can then define several different GSMs including: relative graphlet frequency distance, graphlet degree distribution agreement, and graphlet correlation matrix/distance.

IV. Approach I: Indirect HSMs Based on Graph Representation

The first approach we a propose for defining HSMs is based on transforming the hypergraph into a graph representation and then invoking the standard GSMs.

A. Graph-based Hypergraph Representation

There are two main ways to transform a hypergraph in form of a standard graph: clique expansion and star expansion [13], see Fig. 1. Once hypergraph is represented in form of a standard graph, one can define appropriate adjacency matrix and graph Laplacian. Rather than first transforming hypergraph into a graph, some authors define hypergraph Laplacian directly using analogies from the graph Laplacian. However, it was shown in [13], that several of these direct definitions are special cases of clique or star expansion which follows from different ways of deriving edge weights for the transformed graph from hyperedge weights of the hypergraph. Thus, we focus on clique expansion and star expansion.

a) Clique expansion: The clique expansion algorithm constructs a graph $G^c = (V, E^c \subset V^2, w^c(\cdot))$ from the original hypergraph $\mathcal{G} = (V, E, w(\cdot))$ by replacing each hyperedge $e = (u_1, \ldots, u_{d(e)}) \in E$ with an edge for each pair of vertices in the hyperedge: $E^c = \{(u,v) : u,v \in e, e \in E\}$. Note that the vertices in hyperedge e form a clique in the graph G^c. The edge weight $w^c(u,v)$ can be defined in different ways leading to different clique expansions. Thus, the normalized Laplacian of the constructed graph $\mathcal{G}^c$ becomes

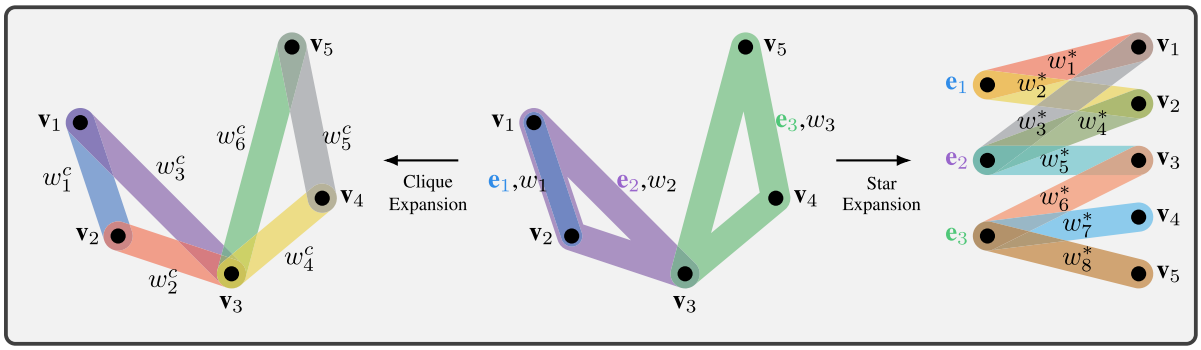

Fig. 1. Illustration of transformation of undirected weighted hypergraph into a graph by clique and star expansion. Also shown are the weights on the edges/ hyperedges.

$$\mathbf{L}^c = \mathbf{I} - (\mathbf{D}_v^c)^{-1/2}\mathbf{A}^c(\mathbf{D}_v^c)^{-1/2},$$

where, $\mathbf{A}^c$ is the adjacency matrix

$$[\mathbf{A}^c]_{uv} = w^c(u,v),$$

and, $\mathbf{D}_v^c$ is the vertex degree matrix with diagonal entries $d^c(u), u \in V$.

The *standard clique* approach minimizes the difference between the edge weight of G^c and the weight of each hyperedge e that contains both u and v leading to,

$$\begin{aligned} w^{cs}(u,v) &= \sum_e h(u,e)h(v,e)w(e), d^{cs}(u) \\ &= \sum_{e\in E} h(u,e)(d(e)-1)w(e), \end{aligned}$$

and thus,

$$\mathbf{A}^{cs} = \mathbf{H}\mathbf{W}\mathbf{H}^T, \tag{10}$$

$$\mathbf{L}^{cs} = \mathbf{I} - (\mathbf{D}_v^{cs})^{-1/2}\mathbf{A}^{cs}(\mathbf{D}_v^{cs})^{-1/2}. \tag{11}$$

Note that the above procedure can still be applied for unweighted hypergraphs by using unit weights, i.e., $w(e) = 1, \forall e \in E$. For unweighted hypergraphs however, one could use other constructions as well, see [13] for details. For example, Bolla [36] proposed to use the weight/adjacency matrix,

$$\mathbf{W}^{co} = \mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^T,$$

leading to the combinatorial Laplacian for G^c being,

$$\mathbf{L}^{co} = \mathbf{D}_v - \mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^T. \tag{12}$$

This definition arises in context of clustering of vertices in a hypergraph, and the eigenvectors of $\mathbf{L}^{co}$ define the "best" Euclidean embedding of the hypergraph in a sense of minimum variance placement, i.e., vertices having many incident edges in common are "closer" to each other in the Euclidean embedding.

b) Star expansion: The star expansion algorithm constructs a graph $G^* = (V^*, E^*, w^*(\cdot))$ from the hypergraph $\mathcal{G} = (V, E, w(\cdot))$ by introducing a new vertex for every hyperedge $e \in E$, thus $V^* = V \cup E$. It connects the new graph vertex e to each vertex in e, i.e. $E^* = \{(u,e) : u \in e, e \in E\}$. Note that each hyperedge in E corresponds to a star in the graph G^*, and G^* is a bi-partite graph. As in clique expansion, different choices can be made for edge weights $w^*(u,e)$ of G^*. In general, the adjacency matrix $\mathbf{A}^*$ of G^* can be expressed as,

$$\mathbf{A}^* = \begin{pmatrix} \mathbf{0}_{|V|} & \mathbf{W}^* \\ (\mathbf{W}^*)^T & \mathbf{0}_{|E|} \end{pmatrix},$$

and the normalized Laplacian can be shown to be,

$$\mathbf{L}^* = \begin{pmatrix} \mathbf{I} & -\mathbf{B}^* \\ -(\mathbf{B}^*)^T & \mathbf{I} \end{pmatrix},$$

where, $\mathbf{B}^*$ is the $|V| \times |E|$ matrix

$$\mathbf{B}^* = (\mathbf{D}_v^*)^{-1/2}\mathbf{W}^*(\mathbf{D}_e^*)^{-1/2},$$

where, $\mathbf{D}_v^*$ and $\mathbf{D}_e^*$ are degree matrices with diagonal entries $d^*(u)$, and $d^*(e)$, respectively, where,

$$\begin{aligned} d^*(u) &= \sum_{e\in E} w^*(u,e), \qquad u \in V, \\ d^*(e) &= \sum_{u\in V} w^*(u,e), \qquad e \in E. \end{aligned}$$

Note that since number of hyperedges, i.e., $|E|$ can be large, the star expansion would result in a graph G^* which can have very large number of vertices making the application of GSM challenging. Furthermore, even if two hypergraphs $\mathcal{G}$ and $\tilde{\mathcal{G}}$ are defined on same node set V to begin with, the star expansions, G^* and $\tilde{G}^*$, respectively will in general have a different set of nodes.

To alleviate these issues, we propose to use the notion of projected Laplacian as defined in [13]. Note that for any $|V| + |E|$ eigenvector $\mathbf{v}^T = [\mathbf{v}_v^T, \mathbf{v}_e^T]$ of $\mathbf{L}^*$ that satisfies $\mathbf{L}^*\mathbf{v} = \lambda\mathbf{v}$, it follows that,

$$\mathbf{B}^*(\mathbf{B}^*)^T\mathbf{v}_v = (\lambda - 1)^2\mathbf{v}_v.$$

Thus, the $|V|$ elements of the eigenvectors of $\mathbf{L}^*$ corresponding to vertices $V \subset V^*$ are eigenvector of the $|V| \times |V|$

matrix,

$$\mathbf{B}^*(\mathbf{B}^*)^T = (\mathbf{D}_v^*)^{-1/2}\mathbf{W}^*(\mathbf{D}_e^*)^{-1}(\mathbf{W}^*)^T(\mathbf{D}_v^*)^{-1/2}.$$

Given this relationship between $\mathbf{B}^*(\mathbf{B}^*)^T$ and $\mathbf{L}^*$,

$$\mathbf{L}_p^* = \mathbf{I} - \mathbf{B}^*(\mathbf{B}^*)^T = \mathbf{I} - (\mathbf{D}_v^*)^{-1/2}\mathbf{A}_p^*(\mathbf{D}_v^*)^{-1/2},$$

can be considered a projected normalized Laplacian on the node set of the original hypergraph $\mathcal{G}$, with $\mathbf{A}_p^*$,

$$\mathbf{A}_p^* = \mathbf{W}^*(\mathbf{D}_e^*)^{-1}(\mathbf{W}^*)^T,$$

being the projected adjacency matrix. Note that the eigenvalues of $\mathbf{L}_p^*$ lie in [0,1], i.e., $0 = \lambda_{1p} \leq \cdots <= \lambda_{np} \leq 1$.

We next discuss different choices for weights $w^*(u,e)$. The *standard star* expansion approach assigns the scaled hyperedge weight, i.e., $w^{*s}(u,e) = \frac{w(e)}{d(e)}$ to each corresponding start graph edge, so that the weight matrix becomes,

$$\mathbf{W}^{*s} = \mathbf{H}\mathbf{W}\mathbf{D}_e^{-1},$$

leading to

$$d^*(u) = \sum_{e\in E} h(u,e)w(e)/\delta(e), \quad u \in V,$$
$$d^*(e) = w(e), \quad e \in E,$$

expressed in terms of original hypergraph's $\mathcal{G}$ properties. Another choice is $w^*(u,e) = w(e)$, i.e.,

$$\mathbf{W}^{*z} = \mathbf{H}\mathbf{W},$$

which leads to $\mathbf{D}_v^* = \mathbf{D}_v$ and $\mathbf{D}_e^* = \mathbf{W}\mathbf{D}_e$, and thus resulting in,

$$\mathbf{A}_p^{*z} = \mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{W}\mathbf{H}^T, \tag{13}$$

and

$$\mathbf{L}_p^{*z} = \mathbf{I} - \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}, \tag{14}$$

which is the same hypergraph Laplacian as the one proposed by Zhuo et al. [37]. This definition of hypergraph Laplacian originates from relaxation of normalized hypergraph cut problem analogous to the standard normalized graph cut problem. Infact, the eigenvector of $\mathbf{L}_p^{*z}$ corresponding to its second smallest eigenvalue encodes the information about subsets of vertices in the hypergraph which are weakly connected to each other.

For transforming hypergraph into graph for similarity comparison using GSMs, we propose to explore both the standard clique expansion ((10) and (11)), and projected star expansion based on Zhuo et al. construction ((13) and (14)).

B. Indirect HSMs

Let $\mathcal{G}$ be a weighted hypergraph, and let G_H be the graph obtained by one of the approaches discussed in the previous section. We will denote this transformation as $G_H = \mathcal{T}(\mathcal{G})$. Given a transformation $\mathcal{T}$ and a GSM D, an indirect HSMs $\mathcal{D}$ induced by the pair $(\mathcal{T}, D)$ is given by,

$$\mathcal{D}_{\mathcal{T},D}(\mathcal{G},\tilde{\mathcal{G}}) \equiv D(\mathcal{T}(\mathcal{G}),\mathcal{T}(\tilde{\mathcal{G}})) = D(G_H,\tilde{G}_H). \tag{15}$$

Note that depending on whether $\mathcal{G}$ and $\tilde{\mathcal{G}}$ have known or unknown node correspondence, appropriate D can be chosen from the GSMs discussed in Section III or any other available in the literature. Furthermore, depending on application one can pick D to capture local, global, mesoscopic, or multi-scale differences.

For numerical demonstration we will choose one representative example of a local, global, mesoscopic, and multiscale HSMs, namely, Hamming HSM (local), spectral HSM (global), eigenvector centrality-based HSM (mesoscopic), and deltaCon HSM (multiscale), see Section VI for details. For these selected HSMs, we summarize in the Table I which SM properties (as listed in Section II-C) are satisfied. This is based on empirical analysis presented in [26], [32], [38], [39]: consequently, the deductions in the table (other than for PI) are limited by the graph topologies considered in the study and may not generalize to all settings. Furthermore note that since in this paper we only consider hypergraphs defined on same node set (and so having same number of nodes), we have not included the size-invariance property in the table.

TABLE I
SOME SELECTED INDIRECT HSMS AND SM PROPERTIES THEY SATISFY

	EI	ES	WA	FA	PI	SA
Hamming			✓			
Spectral			✓	✓	✓	
Centrality	✓		✓	✓		
DeltaCon	✓	✓	✓	✓		✓

If the property is satisfied it is denoted by a check mark, otherwise left empty. For the full forms of the abbreviations for the sm properties, see section II-C.

V. APPROACH II: DIRECT HSMS BASED ON TENSOR-BASED HYPERGRAPH REPRESENTATION

In this section we propose a second approach for defining HSMs which is based on hypergraph representations that intrinsically capture multi-way relations using tensors. Recently, there has been an increasing application of tensors for hypergraph analysis, see for example [17], [40], [41], [42], [43], [44].

A. Tensor Preliminaries

A tensor is a multidimensional array [16], [45], [46], [47]. The order of a tensor is the number of its dimensions, and each dimension is called a mode. An m-th order real valued tensor will be denoted by $\mathsf{X} \in \mathbb{R}^{J_1\times J_2\times\cdots\times J_m}$, where J_k is the size of its k-th mode. We will denote by $\mathcal{J} = (J_1, J_2, \ldots, J_m)$.

The inner product of two tensors $\mathsf{X}, \mathsf{Y} \in \mathbb{R}^{J_1\times J_2\times\cdots\times J_m}$ is defined as,

$$\langle \mathsf{X}, \mathsf{Y}\rangle = \sum_{j_1=1}^{J_1} \cdots \sum_{j_m=1}^{J_m} \mathsf{X}_{j_1 j_2 \ldots j_m}\mathsf{Y}_{j_1 j_2 \ldots j_m},$$

leading to the tensor Frobenius norm $\|\mathsf{X}\|^2 = \langle \mathsf{X}, \mathsf{X}\rangle$. We say two tensors X and Y are orthogonal if the inner product $\langle \mathsf{X}, \mathsf{Y}\rangle = 0$. The matrix tensor multiplication $\mathsf{X} \times_k \mathbf{A}$ along

mode k for a matrix $\mathbf{A} \in \mathbb{R}^{I \times J_k}$ is defined by, $(\mathsf{X} \times_k \mathbf{A})_{j_1 j_2 \cdots j_{k-1} i j_{k+1} \cdots j_m} = \sum_{j_k=1}^{J_k} \mathsf{X}_{j_1 j_2 \cdots j_k \cdots j_m} \mathbf{A}_{i j_k}$. This product can be generalized to what is known as the Tucker product,

$$\mathsf{X} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \cdots \times_m \mathbf{A}_m \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m}. \tag{16}$$

Tensor unfolding is considered as a critical operation in tensor computations [16], [45], [48]. In order to unfold a tensor $\mathsf{X} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_m}$ into a vector or a matrix, we use an index mapping function $ivec(\cdot, \mathcal{J}) : \mathbb{Z}^+ \times \mathbb{Z}^+ \times \overset{m}{\cdots} \times \mathbb{Z}^+ \to \mathbb{Z}^+$ as defined in [48], which is given by

$$ivec(\mathbf{j}, \mathcal{J}) = j_1 + \sum_{k=2}^{m} (j_k - 1) \prod_{l=1}^{k-1} J_l.$$

where, $\mathbf{j} = (j_1, j_2, \ldots, j_m)$. The k-mode unfolding of X denoted by $\mathsf{X}_{(k)}$, is a $J_k \times J_1 \cdots J_{k-1} J_{k+1} \cdots J_m$ matrix, whose (i, p)-th entries are given by

$$\mathsf{X}_{(k)}(i, p) = \mathsf{X}_{j_1 \cdots j_{k-1} i j_{k+1}, \cdots j_m},$$

where, $\tilde{\mathbf{j}} = (j_1, \ldots, j_{k-1}, j_{k+1}, \ldots, j_m)$ is such that $p = ivec(\tilde{\mathbf{j}}, \mathcal{J})$ with $\tilde{\mathcal{J}} = (J_1, \ldots, J_{k-1}, J_{k+1} \cdots J_m)$.

a) HOSVD: Higher-order Singular Value Decomposition (HOSVD) is a multilinear generalization of matrix SVD to tensors [49]. HOSVD of a tensor $\mathsf{X} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_m}$ is given by:

$$\mathsf{X} = \mathsf{S} \times_1 \mathbf{U}_1 \times_2 \cdots \times_m \mathbf{U}_m, \tag{17}$$

where, $\mathbf{U}_k \in \mathbb{R}^{J_k \times R_k}$ satisfies $\mathbf{U}_k^T \mathbf{U}_k = \mathbf{I}$, and $\mathsf{S} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_m}$ is called the core tensor. The quantity $R_k \leq J_k$ is referred to as the k-mode multilinear rank of X, and is equal to rank of k-mode matrix unfolding of X, i.e., $R_k = \mathrm{rank}(\mathsf{X}_{(k)})$. The subtensors $\mathsf{S}_{j_k=\alpha}$ of S obtained by fixing the k-th mode to α, have the properties:

1) all-orthogonality: two subtensors $\mathsf{S}_{j_k=\alpha}$ and $\mathsf{S}_{j_k=\beta}$ are orthogonal for all possible values of k, α and β subject to $\alpha \neq \beta$;
2) ordering: $\|\mathsf{S}_{j_k=1}\| \geq \ldots \geq \|\mathsf{S}_{j_k=J_k}\| \geq 0$ for all possible values of k.

The Frobenius norms $\|\mathsf{S}_{j_k=j}\|$, denoted by $\gamma_j^{(k)}$, are known as the k-mode singular values of X. De Lathauwer et al. [49] showed that the number of nonvanishing k-mode singular values of a tensor is equal to R_k. The classic strategy for computing HOSVD involves a sequence of matrix SVDs:

- For $k = 1, 2, \ldots, m$, do the following:
  - Construct the k-mode matrix unfolding of $\mathsf{X}_{(k)}$,
  - Compute the compact SVD of the matrix $\mathsf{X}_{(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$ and store the left singular vectors $\mathbf{U}_k \in \mathbb{R}^{J_k \times R_k}$,
- Compute the core tensor $\mathsf{S} = \mathsf{X} \times_1 \mathbf{U}_1^T \times_2 \cdots \times_m \mathbf{U}_m^T$.

Other faster methods are available, see for example [50].

b) Tensor Eigenvalues/Eigenvectors: Consider a m-th order n dimensional cubical (i.e. with equal size $J_i = n, i = 1, \ldots, m$ in all modes) tensor $\mathsf{A} \in \mathbb{R}^{n \times n \times \cdots \times n}$. A is called super-symmetric if $\mathsf{A}_{i_1, \ldots, i_m} = \mathsf{A}_{\sigma(i_1 \ldots i_m)}$ for all $\sigma \in \Sigma_m$, the symmetric group of m indices. To a n-vector $\mathbf{x} = (x_1, \ldots, x_n)^T$, real or complex, define a n-vector via Tucker product as:

$$\mathsf{A}\mathbf{x}^{m-1} = \mathsf{A} \times_2 \mathbf{x} \times_3 \cdots \times_m \mathbf{x}.$$

There are many different notions of tensor eigenvalues/eigenvectors [51], [52]. A pair $(\lambda, \mathbf{x}) \in \mathbb{R} \times \{\mathbb{R}^n \setminus \{\mathbf{0}\}\}$ is called

- H-eigenvalue/eigenvector (or H-eigenpair) of A if it satisfies,
$$\mathsf{A}\mathbf{x}^{m-1} = \lambda \mathbf{x}^{[m-1]}, \tag{18}$$
where, $(\mathbf{x}^{[m-1]})_i = x_i^{m-1}$. Eigenvalue λ is called H^+ or H^{++} eigenvalue, if the corresponding eigenvector satisfy $\mathbf{x} \geq 0$, or $\mathbf{x} > 0$, respectively.
- Z-eigenvalue/eigenvector (or Z-eigenpair) of A if it satisfies,
$$\begin{aligned} \mathsf{A}\mathbf{x}^{m-1} &= \lambda \mathbf{x}, \\ x_1^2 + \cdots + x_n^2 &= 1. \end{aligned} \tag{19}$$
- l^p-eigenvalue/eigenvector (or l^p-eigenpair) of A for any $p > 0$ if it satisfies,
$$\begin{aligned} \mathsf{A}\mathbf{x}^{m-1} &= \lambda \mathbf{x}^{[p-1]}, \\ x_1^p + \cdots + x_n^p &= 1. \end{aligned} \tag{20}$$

Note that l^p-eigenvalue/eigenvector reduce to H-eigenvalue/eigenvector and Z-eigenvalue/eigenvector for $p = m$ and $p = 2$, respectively, where note the constraint in (20) is superfluous for $p = m$.

It was proved in [51] that H-eigenvalues and Z-eigenvalues exist for an even order real super-symmetric tensor. A numerical procedure for computing H-eigenvalues is provided in [53] with an associated MATLAB toolbox [54]. The procedure involves homotopy continuation type method which can be computationally intensive, thus making it challenging to scale to large order/size tensors.

B. Tensor-based Hypergraph Representation

We follow tensor-based formulation proposed in [17] to define hypergraph adjacency tensor and Laplacian tensor. Let $\mathcal{G} = (V, E, w(\cdot))$ be a weighted hypergraph with n vertices, and k be the maximum cardinality of the hyperedges, i.e. $k = \max\{|e| : e \in E\}$.

a) Adjacency Tensor: The adjacency tensor $\mathsf{A} \in \mathbb{R}^{n \times n \times \cdots \times n}$ of $\mathcal{G}$, which is a k-th order n-dimensional super-symmetric tensor, is defined as,

$$\mathsf{A}_{j_1 j_2 \cdots j_k} = \begin{cases} \frac{w(e)s}{\alpha} \text{ if } e = (i_1, i_2, \ldots, i_s) \in E \\ \\ 0, \quad \text{otherwise} \end{cases}, \tag{21}$$

where, $j_1, j_2, \ldots, j_k$ are chosen in all possible ways from $\{i_1, i_2, \ldots, i_s\}$ with at least once for each element of the set, and

$$\alpha = \sum_{k_1,\ldots,k_s \geq 1, \sum_{i=1}^{s} k_i = k} \frac{k!}{\prod_{i=1}^{s} k_i!}.$$

Using the adjacency tensor, the degree $d(v_i)$, of a vertex $v_i \in V$, can be expressed as,

$$d(v_i) = \sum_{j_1, j_2, \ldots, j_{k-1}=1}^{n} \mathsf{A}_{i j_1 j_2 \cdots j_{k-1}}. \tag{22}$$

The choice of the nonzero coefficients $\frac{w(e)s}{\alpha}$ preserves the degree of each node, i.e., the degree of node j computed using (22) with weights as defined above is equal to number of hyperedges containing the node in the original non-uniform hypergraph. Note that for a k-uniform hypergraph, above definition simplifies to,

$$\mathsf{A}_{j_1 j_2 \cdots j_k} = \begin{cases} \frac{w(e)}{k-1!} \text{ if } \mathrm{e} = (i_1, i_2, \ldots, i_k) \in E \\ \\ 0, \quad \text{otherwise} \end{cases}. \tag{23}$$

b) Laplacian Tensor: Let D be a k-th order n-dimension super-diagonal tensor with nonzero elements $d_{ii\cdots i} = d(v_i)$. The hypergraph Laplacian tensor is defined as,

$$\mathsf{L} = \mathsf{D} - \mathsf{A}, \tag{24}$$

which is also a k-th order n-dimension super-symmetric tensor. Similarly, normalized hypergraph Laplacian tensor can also be defined. We recall a result from [17], which establishes following properties (which are analogous to case of graph Laplacian) of L,

- L has an H-eigenvalue 0 with eigenvector $\mathbf{v} = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$. Moreover, 0 is the unique H^{++}-eigenvalue of L,
- Δ is the largest H^{+}-eigenvalue of L, where Δ is maximum node degree of $\mathcal{G}$,
- $(d(v_i), \mathbf{e}_i)$ is an H-eigenpair, where $\mathbf{e}_i \in \mathbb{R}^n$ are the standard basis vectors.

H-eigenvalues of L, thus encode global structural properties of a hypergraph, and we propose to use them in generalizing spectral GSM for hypergraphs.

c) Example: To motivate the advantage of using tensor-based representation of hypergraphs over graph based representation, we consider an example. Let $V = \{1, 2, \ldots, 6\}$, and let $\mathcal{G} = (V, E)$ and $\tilde{\mathcal{G}} = (V, \tilde{E})$ be two hypergraphs, with hyperedges $E = \{\{1, 2, 3\}, \{1, 5, 6\}, \{3, 4, 5\}, \{2, 4, 6\}\}$ and $\tilde{E} = \{\{1, 2, 6\}, \{1, 3, 5\}, \{2, 3, 4\}, \{4, 5, 6\}\}$, respectively. It can be shown that $\mathcal{G}$ and $\tilde{\mathcal{G}}$ are non-isomorphic (i.e. no permutation of nodes will makes two hypergraphs same), but lead to same clique and star expansion. On the other hand the adjacency tensor A and $\tilde{\mathsf{A}}$ associated with $\mathcal{G}$ and $\tilde{\mathcal{G}}$, respectively are different, i.e., $\mathsf{A} \neq \tilde{\mathsf{A}}$. Hence, while indirect HSM based on clique expansion will results in a zero value, direct HSM based on tensor representation (see Section V) will result in a non-zero value. However, computing direct HSMs can be computationally challenging. A detailed discussion of pros/cons of direct and indirect HSMs is given in the Section VI-D.

d) HOSVD and Centrality for Hypergraphs: We next discuss HOSVD of L and associated properties. Since L is super-symmetric, any mode unfolding of L would yield the same unfolding matrix with the same singular values which we denote by $\gamma_j, j = 1, \ldots, n$ (note that we have removed dependence of γ_j^k on mode k). It was shown in [18] that the singular values of L encode structural properties of the hypergraph, such as vertex degrees, path lengths, clustering coefficients and nontrivial symmetricity for uniform hypergraphs, and thus can be used to quantify differences in hypergraph structure. Moreover, a fast and memory efficient tensor train decomposition (TTD)-based computational framework was developed in [18] to compute the singular values for uniform hypergraphs. Given these two desirable features, we also propose to use singular values of L as an alternative to H-eigenvalues in defining spectral HSM.

The notion of centrality has been generalized for hypergraphs. H/Z-eigenvectors of the adjacency tensor A are used to define hypergraph eigenvector-centrality [40]. In particular, the H/Z-eigenvector centrality $\mathbf{c} \in \mathbb{R}^n$ is defined to be an H/Z-eigenvector of the adjacency tensor A such that it is positive i.e., $\mathbf{c} > 0$ with a positive H/Z-eigenvalue, i.e., $\lambda > 0$. By Perron-Frobenius theorem for non-negative tensors [55], such positive H/Z-eigenvectors exist under certain irreducibility conditions on A. While such a positive Z-eigenpair may not be unique, the H-eigenpair is always unique up to scaling. Along similar lines, authors in [56] define node and hyperedge centralities as vectors $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{e} \in \mathbb{R}^m$, respectively, that satisfy,

$$\mathbf{c}\lambda = g(\mathbf{H}\mathbf{W}f(\mathbf{e})), \tag{25}$$

$$\mathbf{e}\mu = \psi(\mathbf{H}^T\phi(\mathbf{c})), \tag{26}$$

$$\text{s.t.} \quad c, \mathbf{e} > 0, \lambda, \mu > 0. \tag{27}$$

In above system of equations, $\mathbf{H}$ is the incidence matrix and $\mathbf{W}$ is the hyperedge weight matrix for $\mathcal{G}$ (and we have assumed that all vertices have unit weight consistent with setting in this paper), and $g, f, \phi, \psi : \mathbb{R}^+ \to \mathbb{R}^+$ are appropriately chosen non-negative functions on non-negative real domain. Furthermore note that these scalar functions are extended on vectors by defining them as mappings that act in a componentwise fashion. By invoking Perron-Frobenius theorem for multi-homogeneous mappings [57], it was proved that under certain conditions on the scalar functions, the solution of above system exists and is unique. A nonlinear power method with convergence guarantees is proposed to solve the above system. Furthermore, it is shown that with choices $f(\mathbf{x}) = \mathbf{x}, g(\mathbf{x}) = \mathbf{x}^{1/(p+1)}, \psi(\mathbf{x}) = e^{\mathbf{x}}$ and $\psi(\mathbf{x}) = \ln(\mathbf{x})$, the node centrality vector $\mathbf{c}$ is also a l^p-tensor eigenvector, and thus further generalizing the notion of H/Z-eigenvector centrality.

C. Direct HSMs

For tensor-based representation we define a set of SMs along similar lines as discussed in Section III. Let $\mathcal{G}$ and $\tilde{\mathcal{G}}$ be two weighted hypergraphs with same node set and same

maximum hyperedge cardinality, and let (A, L) and $(\tilde{\mathsf{A}}, \tilde{\mathsf{L}})$ be corresponding adjacency and Laplacian tensor, respectively.

- *Structural HSM:* It is straightforward to generalize the Hamming and Jaccard distance for graphs to tensor-based representation as follows:

$$\mathcal{D}_H(\mathcal{G}, \tilde{\mathcal{G}}) = \frac{||\mathsf{A} - \tilde{\mathsf{A}}||_1}{\mathcal{N}},$$

where,

$$||\mathsf{A}||_1 = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_k=1}^{n} |\mathsf{A}_{i_1,\ldots,i_k}|,$$

is tensor 1-norm and $\mathcal{N} = n^k - n$ is a normalization constant, and

$$\mathcal{D}_J(\mathcal{G}, \tilde{\mathcal{G}}) = 1 - \frac{\sum_{j_1 j_2 \ldots j_k} \min(\mathsf{A}_{j_1 j_2 \ldots j_k}, \tilde{\mathsf{A}}_{j_1 j_2 \ldots j_k})}{\sum_{j_1 j_2 \ldots j_k} \max(\mathsf{A}_{j_1 j_2 \ldots j_k}, \tilde{\mathsf{A}}_{j_1 j_2 \ldots j_k})},$$

respectively.

- *Feature-based HSM:* As in feature-based GSM, we can use specific "features" of the hypergraph, such as the node degree distribution, different notions of centrality, diameter, etc for use in comparing hypergraphs. If we are using node dependent features, the method aggregates a feature-vertex matrix of size $k \times n$, where k is number of features selected. This feature-vertex matrix for the two hypergraphs can then be directly compared, or can be further reduced to a "signature vector" as in the graph case, and used to obtain a SM between hypergraphs. As in graph case (see Section III) we propose to use tensor-based hypergraph centrality as the feature for comparison. Let $c_i, i = 1, \ldots, n$ and $\tilde{c}_i, i = 1, \ldots, n$ be normalized (i.e., $|\mathbf{c}|_1 = |\tilde{\mathbf{c}}|_1 = 1$) node centralities for $\mathcal{G}$ and $\tilde{\mathcal{G}}$, respectively, then centrality-based HSM is given by,

$$\mathcal{D}_C(\mathcal{G}, \tilde{\mathcal{G}}) = \frac{1}{n} \sum_{i=1}^{n} |c_i - \tilde{c}_i|. \tag{28}$$

While in above definition one could use any notion of hypergraph centrality, we propose to use the node centrality defined by (25)-(27) in our application.

- *Spectral HSM:* Let the ordered set of H-eigenvalues of L be $\lambda_1, \ldots, \lambda_p$ i.e., $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_q$, and similarly let $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{\tilde{q}}$ be the ordered set for $\tilde{\mathsf{L}}$. Note that in general $q \neq \tilde{q}$. Without loss of generality, assume $\tilde{q} > q$ and define an extended set of H-eigenvalues $\overline{\lambda}_1, \ldots, \overline{\lambda}_{\tilde{q}}$ for $\mathcal{G}$, where $\overline{\lambda}_i = 0, i \leq \tilde{q} - q$, and $\overline{\lambda}_i = \lambda_{i-(\tilde{q}-q)}, i = \tilde{q} - q + 1, \ldots, \tilde{q}$. The l_p distance on space of H-eigenvalues can then be defined as,

$$\mathcal{D}_\lambda(\mathcal{G}, \tilde{\mathcal{G}}) = \frac{1}{\tilde{q}} \sum_{i=1}^{\tilde{q}-1} |\lambda_i - \tilde{\lambda}_i|^p. \tag{29}$$

As discussed above, we similarly propose to use higher-order singular values, leading to,

$$\mathcal{D}_\gamma(\mathcal{G}, \tilde{\mathcal{G}}) = \frac{1}{n} \sum_{i=1}^{n-1} |\gamma_i - \tilde{\gamma}_i|^p, \tag{30}$$

where, $\gamma_i, i = 1, \ldots, n$ and $\tilde{\gamma}_i, i = 1, \ldots, n$ are higher-order singular values of L and $\tilde{\mathsf{L}}$, respectively. We will refer to $\mathcal{D}_\lambda$ and $\mathcal{D}_\gamma$ as spectral-H and spectral-S HSMs, respectively.

- *Hypergraph Embedding-based HSMs:* Recently, GNNs have been extended to hypergraph neural networks (HNN) [58], [59]. Thus, as for graphs (see Section III), one can learn an embedding from a set of hypergraphs into an Euclidean space, and then compute a distance between the embeddings of two hypergraphs.
- *Hypergraph Kernel-based HSMs:* The notion of graph kernels has been generalized to hypergraphs, see for example [60], [61]. These kernels can be used for comparing hypergraphs as in the graph case.

Given the similarity in the construction to their graph counterparts, the SM properties (as discussed in Section II-C) satisfied by Hamming, spectral, and centrality-based direct HSMs mentioned above are expected to be similar to their graph counterparts, see Table I for details. However, further empirical analysis may be necessary to confirm this assertion.

VI. Numerical Studies

In this section we assess the performance of indirect and direct HSMs on synthetic hypergraphs and real-world biological datasets. For these studies we have chosen one representative example of a local, global, mesoscopic, and multiscale HSMs, namely, Hamming HSM (local), spectral HSM (global), centrality-based HSM (mesoscopic), and deltaCon HSM (multiscale).

A. Synthetic Hypergraphs

To generate synthetic hypergraphs, we consider three families of generative models: Erdös-Rényi (ER), Barabási-Albert (BA) and Watts-Strogats (WS). These three models are widely used as test-beds in a variety of network science problems and have varying structural complexity. ER model [62] leads to "structureless" graph in the sense that the statistical properties of each edge and vertex in the graph is exactly same. In BA or scale-free (SF) model [63], on the other hand, the node degree distribution behaves as a power-law due to preferential attachment, and that impacts both its global and local structure. On the local scale, vertices in graph tend to connect exclusively to highest-degree vertices in the graph, rather than to one another, generating a tree-like topology. The high-degree vertices acts like hub which are by definition are global structures as they touch a significant portion of rest of the graph, thereby increasing the connectivity throughout the graph. WS model [64] on a global scale looks like an uncorrelated random graph in which it exhibits no communities or high-degree vertices but has small average shortest path length between vertices, while at local scale it shows high clustering compared to the BA model.

The three models, originally developed for graphs, have been generalized to the hypergraph case. We will restrict to procedure of construction of k-uniform hypergraphs in each family. The user specifies the desired number of vertices n, desired number of hyperedges m and some additional parameters depending on the model as discussed below.

a) Erdos–Renyi (ERH): There are $\binom{n}{k}$ possible hyperedges in a k-uniform hypergraph. To construct a random k-uniform hypergraph, we uniformly sample m hyperedges from this set without repetition.

b) Scale Free Hypergraph (SFH(μ)): To construct an k-uniform SFH, we follow the generative model from [65]:

i. Assign each node a probability p_i as:

$$p_i = \frac{i^{-\mu}}{\sum_{j=1}^{n} j^{-\mu}}, \quad i = 1, \ldots, n,$$

where, $0 < \mu < 1$ is a user chosen parameter.

ii. Select k-distinct vertices with probabilities $p_{i_1}, \ldots, p_{i_k}$. If the hypergraph does not already contain a hyperedge of those chosen k vertices, then add the hyperedge to the hypergraph

iii. Repeat step ii) until m unique hyperedges have been generated.

Similar to BA/SF graph, this procedure produces a hypergraph with vertices with average degrees $< d >$ having a power-law distribution, $\mathcal{P}_k(< d >) \sim < d >^{-\lambda}$ with $\lambda = 1 + \frac{1}{\mu}$.

c) Watts-Strogats Hypergraph (WSH(p)): Using a procedure similar for WS graphs, WSH is constructed as follows, [18]:

i. Construct a d-regular k-uniform hypergraph with n vertices, and add extra hyperedges in every $k+1$ vertices. We refer to hyperedges in this hypergraph as the initial hyperedges.

ii. Select an initial hyperedge and generate a new hyperedge with k vertices chosen uniformly at random. If the new hyperedge does not exist, with probability p, replace the selected hyperedge with the new hyperedge. Here $0 < p < 1$ is a rewiring probability as specified by the user.

iii. Repeat step ii), till all initial hyperedges have been iterated on.

Fig. 2, show a realization of each of these three hypergraph models with $k = 4$, $n = 80$ and $m = 100$. To asses if these hypergraph models posses similar structural properties as their graph counterparts, we compare their node degree distribution, average path length, and clustering coefficient. For computing average path length, we use:

$$L_{\mathrm{a}} = \frac{1}{n(n-1)} \sum_{j \neq i} d(v_j, v_i), \tag{31}$$

where, $d(v_j, v_i)$ denotes the shortest distance between vertices v_j and v_i. Note in computing shortest distance, two vertices are considered adjacent if they share a common hyperedge. For clustering coefficient we use definition from [18],

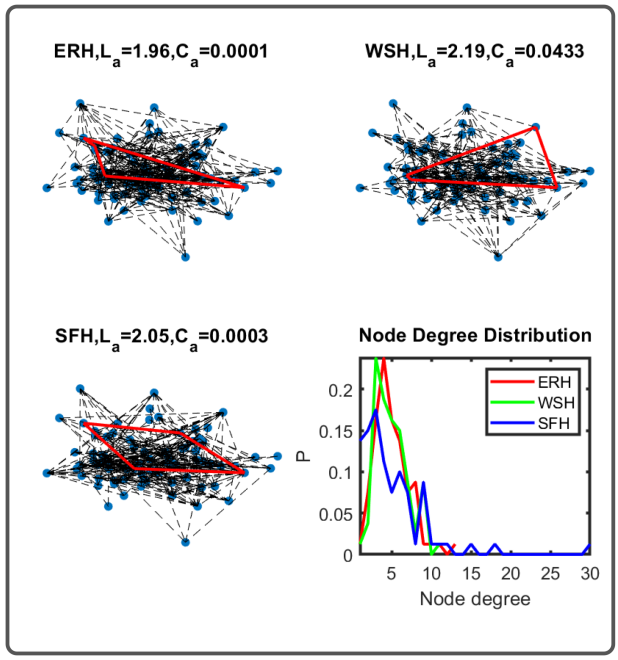

Fig. 2. Examples of 4-uniform ERH, WSH and SFH with $n = 80$ and $m = 100$. Also shown are node degree distribution, average path length and clustering coefficients for each case. The red polygons represent a selected hyperedge.

$$C_j = \frac{|\{e_{i_1 i_2 \cdots i_k} : v_{i_1}, v_{i_2}, \ldots, v_{i_k} \in V_j, e_{i_1 i_2 \cdots i_k} \in E\}|}{\binom{|V_j|}{k}},$$

$$\Rightarrow C_{\mathrm{a}} = \frac{1}{n} \sum_{j=1}^{n} C_j, \tag{32}$$

where, V_j is the set of vertices that are immediately connected to v_j i.e. share a hyperedge with node v_j, and $\binom{|V_j|}{k} = \frac{|V_j|!}{(|V_j|-k)!k!}$ returns the binomial coefficients. If $|V_j| < k$, we set $C_j = 0$.

As can be seen from Fig. 2, the ERH and WSH construction results in a hypergraph with almost homogeneous degree distribution. The SFH, on the other hand, shows power-law degree distribution for node degrees, as discussed above. The WSH model shows high clustering coefficient compared to ERH and SFH as expected.

B. Performance on Synthetic Hypergraphs

We next assess and compare the effectiveness of different HSMs in differentiating hypergraphs with distinct structural features, i.e., originating from the different models. In other words, to yield a good performance, a HSM should be able to assign small distance to hypergraph pairs coming from the same model but large HSM values to pairs coming from different models.

A systematic approach for quantification of the performance can be accomplished via the receiver operating characteristic (ROC) curve [66]. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. For a given HSM, one defines a threshold $\epsilon > 0$ and classifies two hypergraphs as belonging

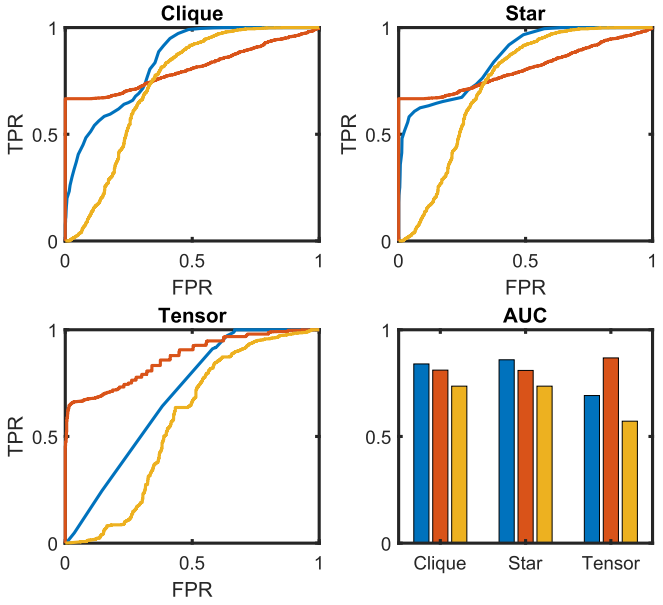

Fig. 3. ROC curves and AUC for different HMDs: Hamming (blue), Spectral (red) and Centrality (orange).

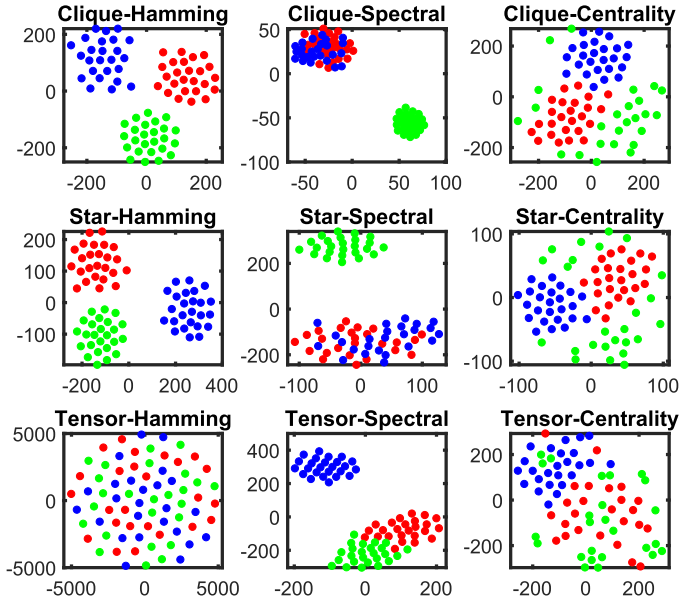

Fig. 4. Embedding for different hypergraphs in 2d using tSNE applied to HSMs: ERH (red), WSH (green) and SFH (blue).

to the same model class if their HSM value is less than ϵ. Given that the correct classes are known, TPR and FPR values can be computed to quantify the accuracy of classifying all the hypergraphs. The procedure is then repeated by varying ϵ, obtaining the ROC curve which, ideally, should have TPR equal to 1 for any FPR value. Furthermore, one can compute the area under curve (AUC) which is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. Thus, for a perfect classifier, AUC $= 1$, while for a classifier that randomly assigns observations to classes, AUC$= 0.5$.

For testing purposes, we consider hypergraph size $n = 40$, $m = 50$ and generate 25 networks for each of the three types, resulting in a total population of $N_g = 75$ hypergraphs. We then compute all the pairwise HSM using each method, ending up with a $N_g \times N_g$ distance matrix for each case. We then generate ROC curves for each HSM considered using procedure discussed above. In addition, to create a visual aid, we also generate a 2d embedding for each hypergraph in the population by applying t-distributed stochastic neighbor embedding (tSNE) [67] to the distance matrix corresponding to each HSM. Fig. 3 shows the ROC curves, while the 2-dimensional embedding is shown in Fig. 4, where we have labeled each point corresponding to different instances of hypergraph using distinct colors based on its known model class. Note that for an unweighted uniform hypergraph, the adjacency matrices for clique (10) and star expansion (13) become same up to a scale factor. Hence, we find that the ROC curves for Hamming and centrality-based indirect HSMs are similar.

C. Test on Real Datasets

In this section we assess the performance of HSMs in grouping sets of networks derived from noisy experimental data. In order to assess statistical significance while comparing two hypergraphs, we propose to use the permutation test.

1) Permutation Test for HSM: Consider a hypothesis testing problem:

- Null (H_0): $\mathcal{G}_1$ and $\mathcal{G}_2$ are similar,
- Alternative (H_1): $\mathcal{G}_1$ and $\mathcal{G}_2$ are dissimilar.

Since the null distribution for H_0 is unknown, we use a permutation test [68] to empirically estimate it. Let $\mathcal{D}$ be any of the HSMs, and let p_s be the desired significance level of the test. The steps in the permutation test involve:

- Step 1: Randomly generate a family of hypergraphs $\{\mathcal{G}_i^r\}_{i=1}^N$ which are similar to $\mathcal{G}_1$, and compute $d_i = \mathcal{D}(\mathcal{G}_1, \mathcal{G}_i^r), i = 1, \ldots, N$.
- Step 2: Compute $d_{12} = \mathcal{D}(\mathcal{G}_1, \mathcal{G}_2)$.
- Step 3: Compute p-value as $p = \frac{1}{N}\sum_{i=1}^N \mathcal{I}_{d_{12}}(d_i)$, where $\mathcal{I}_z$ is indicator function, i.e. $\mathcal{I}_z(x) = 1$ if $x > z$ and $\mathcal{I}_z(x) = 0$ otherwise.
- Step 4: Reject H_0 if $p \leq p_s$.

In Step 1 one could use Erdos-Reyni (ER) or Chung-Lu (CL) procedure [14] to randomly generate hypergraphs with similar characteristics as $\mathcal{G}_1$. Let $\mathbf{d}_v^1 = (d(v_1), \ldots, d(v_n))$ and $\mathbf{d}_e^1 = (d(e_1), \ldots, d(e_m))$ be vertex degree and hyperedge size distribution vectors of $\mathcal{G}_1$. Let $c = \sum_{i=1}^n d(v_i) = \sum_{i=1}^m d(e_i)$, and vertex-hyperedge membership probability in $\mathcal{G}_1$ be p, where

$$p = \frac{c}{mn}.$$

ER procedure selects vertices uniformly at random for each hyperedge with probability p. Thus, for each of the nm vertex-hyperedge pairs, the probability of membership is the same, i.e.,

$$\mathcal{P}(u \in e) = p.$$

On the other hand, the CL procedure generates $\mathcal{G}^r$ with similar vertex degree and hyperedge size distribution as of $\mathcal{G}_1$. The probability a vertex belongs to a hyperedge in $\mathcal{G}^r$ is

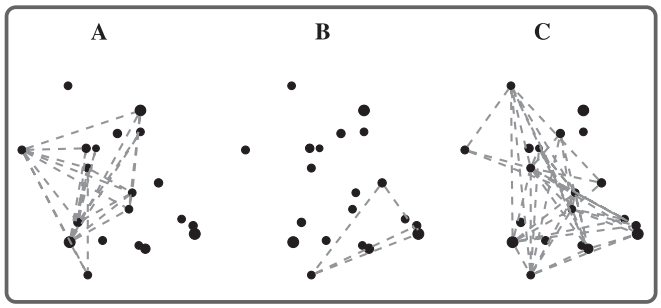

Fig. 5. Mouse neuron endomicroscopy features. (A), (B) and (C) Neuronal activity networks of the three phases - fed, fast and re-fed, which depicts the spatial location and size of individual neurons. Each 2-simplex (i.e., a triangle) represents a hyperedge. The cutoff threshold is 0.93 for the hypergraph model.

proportional to the product of the desired vertex degree and hyperedge size, i.e.,

$$\mathcal{P}(u \in e) = \frac{d(u)d(e)}{c}.$$

To ensure this probability is always less than 1, one may further require the input sequences satisfy $\max_{i,j} d(u_i)d(e_j) \leq c$. Note that this procedure will in general produce a non-uniform hypergraph depending on distribution $\mathbf{d}_e^1$. For a k-uniform hypergraph, $d(e_i) = k, i = 1, \ldots, m$. To sample a k-uniform hypergraph with given node degree distribution $\mathbf{d}_v^1$, we modify the CL process as follows:

i. Assign each node a probability p_i as:

$$p_i = \frac{d(v_i)}{c}, \quad i = 1, \ldots, n.$$

ii. Select k-distinct vertices with probabilities $p_{i_1}, \ldots, p_{i_k}$. If the hypergraph does not already contain a hyperedge of these chosen k vertices, then add the hyperedge to the hypergraph.

iii. Repeat step ii) until m unique edges have been generated.

2) Mouse Neuron Endomicroscopy: The mouse endomicroscopy dataset is an imaging video created under 10-minute periods of feeding, fasting and re-feeding using fluorescence across space and time in a mouse hypothalamus [4], [18], [19]. Twenty neurons are recorded with individual levels of "firing". Similar to [18], we want to quantitatively differentiate the three phases. First, we compute the multi-correlation among every three neurons, which is defined by

$$\rho = (1 - \det(\mathbf{R}))^{\frac{1}{2}}, \tag{33}$$

where, $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is the correlation matrix of three neuron activity levels [69]. When the multi-correlation ρ is greater than a prescribed threshold, we build a hyperedge among the three neurons and assign it an hyperedge weight equal to ρ. For our application we use a threshold of 0.93 as prescribed in [18]. Fig. 5 shows the resulting hypergraphs for the three phases.

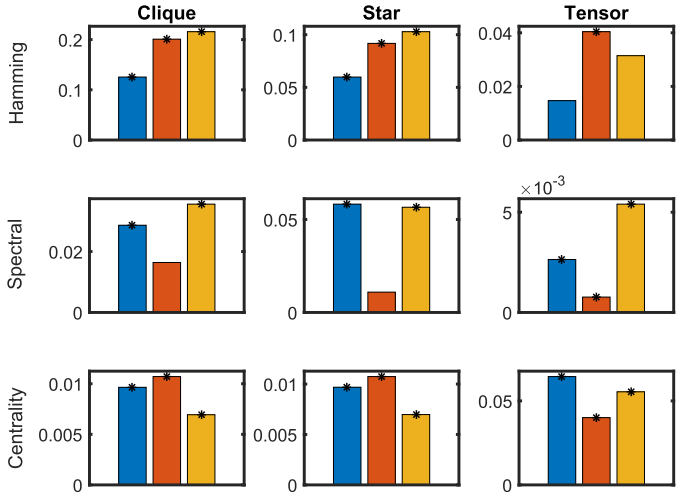

Fig. 6. Comparison of different phases of mouse feeding activity using different HSMs: blue (fed,fast), red (fed,re-fed), and orange (re-fed,fast). The * on the bars implies that there is statistically significant difference between the hypergraphs in the corresponding two phases based on p-values from the permutation test.

Fig. 6 shows the comparison of hypergraph corresponding to different phases using various HSMs. We find similar trends using both indirect and direct HSMs. The spectral and centrality HSMs between fed and re-fed phases have smaller values revealing more similarity at global and mesoscopic scales compared to corresponding HSM values between fed and fast, and re-fed and fast phases. On the other hand, Hamming HSM reveals that fed and fast phase are more similar at local scale compared to fed and re-fed phases. The * on the bars implies that there is statistically significant difference between the hypergraphs in the corresponding two phases based on p-values from the permutation test.

3) Genomic Dataset: We next apply HSM framework to compare genomic structure of two cell types. Genomic DNA must be folded to fit inside a nucleus, but must remain accessible for gene transcription, replication and repair [70], [71]. Consequently, higher-order chromatin structure arises from such combinatorial physical interactions of many genomic loci. Recently, authors in [72] proposed to represent such higher-order chromatin structure by a hypergraph, where the different loci in the genome are the vertices, and each multi-way contact between a set of loci represent a hyperedge. Furthermore, they used Pore-C [73], a recent method developed by Oxford Nanopore Technologies to measure these multi-way contacts directly and construct the hypergraph experimentally. Note that, while Pore-C gives contact information at the finest level of base-pair position in the genome, it is often convenient to aggregate this information at a coarser resolution by aggregating linear continuous segments in the genome, see [72] for details.

Fig. 7 shows the visualization of the incidence matrix of the hypergraph derived for human fibroblasts (FB) and B lymphocytes (GM) cell lines at 25Mb resolution after noise reduction. The entire genome at this resolution consists of

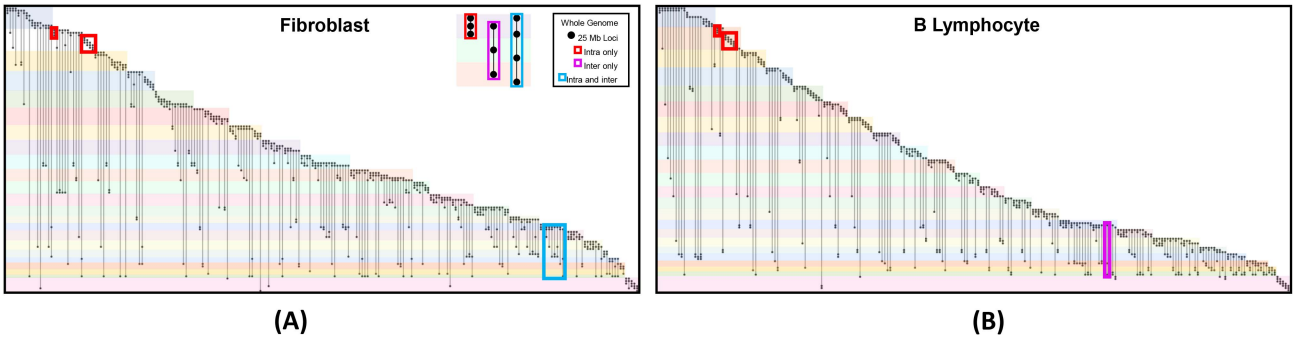

Fig. 7. (A) Incidence matrix visualization of the top 10 most common multi-way contacts per chromosome, for fibroblasts (FB). Matrices are constructed at 25 Mb resolution. Highlighted boxes indicate example intra-chromosomal contacts (red), inter-chromosomal contacts (magenta), and combinations of intra- and inter-chromosomal contacts (blue). Examples for each type of contact are shown in the top right corner. (B) Similar plot for B lymphocytes (GM). Genomic loci that do not participate in the top 10 most common multi-way contacts for fibroblasts or B lymphocytes were removed from these incidence plots.

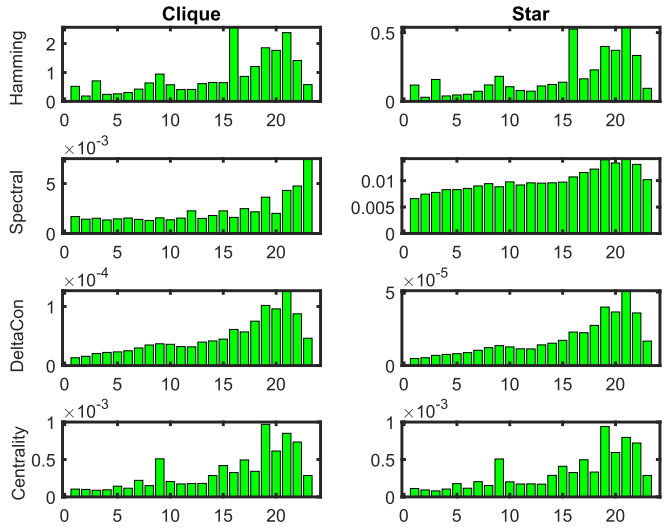

Fig. 8. Comparison of different chromosomes in FB and GM using indirect HSMs.

TABLE II
INDIRECT HSMS VALUES BETWEEN FULL GENOME OF THE FB AND GM

	Clique	Star
Hamming	$2.3 \times 10^{-2}(*)$	$2 \times 10^{-3}(*)$
Spectral	$6.2 \times 10^{-4}(*)$	$1.6 \times 10^{-3}(*)$
deltaCon	$4.7 \times 10^{-7}(*)$	$1.6 \times 10^{-7}(*)$
Centrality	$2.5 \times 10^{-6}(*)$	$2.9 \times 10^{-6}(*)$

*The * in bracket implies that difference between FB and GM is statistically significant based on p-values from the permutation test.*

$n = 3,102$ vertices for both cells, with number of hyperedges $m = 836,571$ for FB and $m = 1,028,694$ for GM. The maximum hyperedge set cardinality is 40 for FB and 90 for GM. To compare chromosomes individually, we also construct separate hypergraphs for each chromosome comprising of intra-chromosomal contacts only. Chromosome 1 has maximum number of vertices $n = 249$ and chromosome 22 has smallest number of vertices $n = 51$ at the chosen 25Mb scale. The number of hyperedges differ by chromosomes taking values in range $[1,000 \quad 35,000]$ for FB, and in the range $[6,000 \quad 75,000]$ for GM, respectively. Moreover maximum hyperedge set cardinality also differs between corresponding chromosomes in the GM and FB. As a result we cannot use the tensor-based direct HSMs, and restrict to indirect HSM for comparison.

In Fig. 8 we show chromosome level comparison using indirect HSM based on the clique and star expansion. We find that the trends between two expansions are similar for Hamming, deltaCon, and centrality HSMs, while they differ for spectral HSM. Furthermore, we can see that at local scale chromosome 16 differs most between FB and GM, while chromosome 19 and 21 differ the most at the mesoscale. The clique-spectral HSM reveals that chromosome 23 differs most at global scale, while star-spectral HSM indicates chromosomes 19,20,21,22 are the most different.

Table II shows the values of different indirect HSMs between FB and GM for the entire genome. The p-values indicate that FB and GM are dissimilar at all scales based on both clique and star expansion.

D. Discussion

We first discuss pros/cons of indirect and direct HSMs. While indirect HSMs allow one to leverage large variety of GSMs for hypergraph comparison, the hypergraph conversion into clique/star representation is lossy (as shown in Example in Section V-B) and thus may result in inability to discern certain aspects of structural differences or similarities between two hypergraphs. Identifying under what conditions (e.g., class of hypergraphs) and how typically such scenarios may arise is an interesting avenue for future research. Direct HSMs being based on tensor representation do not suffer from such loss of information. However, tensor computations (e.g., tensor eigenvalue/singular values) can be challenging for hypergraphs with large number of vertices and/or with high maximum hyperedge cardinality. Moreover, direct HSMs can only be applied to cases where the underlying hypergraph has same number of vertices and same maximum hyperedge cardinality. Indirect HSMs however are computationally less

demanding, and can be employed even if hypergraphs have different maximum hyperedge cardinality. Moreover, by restricting to GSMs which are applicable for comparing graphs with different number of vertices or unknown node correspondence, indirect HSMs can also be applied for comparing hypergraphs with different number of vertices and/or unknown node correspondence. In terms of performance of indirect and direct HSMs to assess structural differences or similarities between two hypergraphs, the numerical studies show that both approaches could be effective depending on the application.

We are currently exploring the application of line expansion [74] which has been recently proposed as an alternative approach to transforming hypergraph into a graph. Compared to clique or star expansion, line expansion does not result in any information loss during the transformation, thus, potentially providing more effective means for developing indirect HSMs. Addressing the computational challenges associated with tensor-based HSM will be important to address to scale the approach to larger problems. In addition approaches alternative to using tensor-based representation, such as higher-order random walk-based hypergraph analysis [14] provide another potential avenue for developing new HSMs.

GNNs have achieved state of art results in graph analytics applications such as node classification and link prediction [75]. As discussed in the Section III, GNNs have also been applied for graph comparison [34] with promising results. It would be worthwhile to further explore application of GNNs as indirect HSMs for hypergraph comparison. Along similar lines, while notions of graph kernels, graph embedding and GNNs have been extended to hypergraphs [58], [59], [60], [61], further investigation is warranted for their application in hypergraph comparison.

In addition use of HSMs in other graph/data analytics problems such as knowledge graph representation learning [76], multiview clustering [77] and node classification/ link prediction [78] would also be worth exploring. Applications of the proposed HSMs in other domains e.g., cybersecurity and social networks is another avenue for future research.

VII. Conclusion

In this article we presented two approaches for hypergraph comparison. The first approach transforms the hypergraph into a graph representation, and then uses standard graph similarity measures. The second approach uses tensors to represent hypergraphs and then invokes various tensor algebraic notions to develop hypergraph similarity measures. Within each approach we presented a collection of measures which assess hypergraph similarity at different scales. We evaluated these measures on synthetic hypergraphs and real-world biological datasets with promising results. Finally, we discussed various pros/cons in using the two approaches, and outlined some avenues of future research.

Acknowledgment

Any opinions, finding, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

[1] M. Newman, *Networks*. Oxford, U.K.: Oxford Univ. Press, 2018.
[2] F. Battiston et al., "Networks beyond pairwise interactions: Structure and dynamics," *Phys. Rep.*, vol. 874, pp. 1–92, 2020.
[3] A. R. Benson, D. F. Gleich, and D. J. Higham, "Higher-order network analysis takes off, fueled by classical ideas and new data," 2021, *arXiv:2103.05031*.
[4] P. Sweeney, C. Chen, I. Rajapakse, and R. D. Cone, "Network dynamics of hypothalamic feeding neurons," *Proc. Nat. Acad. Sci.*, vol. 118, no. 14, 2021, Art. no. e2011140118.
[5] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1738–1745.
[6] D. Di et al., "Hypergraph learning for identification of COVID-19 with CT imaging," *Med. Image Anal.*, vol. 68, 2021, Art. no. 101910.
[7] Q. Fang, J. Sang, C. Xu, and Y. Rui, "Topic-sensitive influencer mining in interest-based social media networks via hypergraph learning," *IEEE Trans. Multimedia*, vol. 16, pp. 796–812, 2014.
[8] Y. Wang, X. Lin, Q. Zhang, and L. Wu, "Shifting hypergraphs by probabilistic voting," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, Springer, 2014, pp. 234–246.
[9] Y. Wang, X. Lin, L. Wu, Q. Zhang, and W. Zhang, "Shifting multi-hypergraphs via collaborative probabilistic voting," *Knowl. Inf. Syst.*, vol. 46, no. 3, pp. 515–536, 2016.
[10] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, "Hypergraph learning: Methods and practices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2548–2566, May 2022.
[11] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*, vol. 45. Amsterdam, Netherlands: Elsevier, 1984.
[12] 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.
[13] S. Agarwal, K. Branson, and S. Belongie, "Higher order learning with graphs," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 17–24.
[14] S. G. Aksoy, C. Joslyn, C. O. Marrero, B. Praggastis, and E. Purvine, "Hypernetwork science via high-order hypergraph walks," *EPJ Data Sci.*, vol. 9, no. 1, pp. 1–34, 2020.
[15] T. Carletti, D. Fanelli, and S. Nicoletti, "Dynamical systems on hypergraphs," *J. Phys.: Complexity*, vol. 1, no. 3, 2020, Art. no. 035006.
[16] T. G. Kolda, "Multilinear operators for higher-order decompositions," Sandia National Laboratories, Livermore, CA, USA, Tech. Rep. SAND2006-2081, 2006.
[17] A. Banerjee, A. Char, and B. Mondal, "Spectra of general hypergraphs," *Linear Algebra Appl.*, vol. 518, pp. 14–30, 2017.
[18] C. Chen and I. Rajapakse, "Tensor entropy for uniform hypergraphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2889–2900, Oct.–Dec. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9119161
[19] C. Chen, A. Surana, A. Bloch, and I. Rajapakse, "Controllability of hypergraphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1646–1657, Apr.–Jun. 2021.
[20] C. Donnat and S. Holmes, "Tracking network dynamics: A survey using graph distances," *Ann. Appl. Statist.*, vol. 12, no. 2, pp. 971–1012, 2018. [Online]. Available: https://doi.org/10.1214/18-AOAS1176
[21] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1365–1374.
[22] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature Commun.*, vol. 6, no. 1, pp. 1–9, 2015.
[23] D. R. Hunter, S. M. Goodreau, and M. S. Handcock, "Goodness of fit of social network models," *J. Amer. Stat. Assoc.*, vol. 103, no. 481, pp. 248–258, 2008.
[24] K. Faust and J. Skvoretz, "Comparing networks across space and time, size and species," *Sociol. Methodol.*, vol. 32, no. 1, pp. 267–299, 2002.
[25] P. Wills and F. G. Meyer, "Metrics for graph comparison: A practitioner's guide," *PLoS One*, vol. 15, no. 2, 2020, Art. no. e0228728.

[26] D. Koutra, J. T. Vogelstein, and C. Faloutsos, "DeltaCon: A principled massive-graph similarity function," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 162–170.
[27] A. Tsitsulin, D. Mottin, P. Karras, A. Bronstein, and E. Müller, "NetLSD: Hearing the shape of a graph," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2347–2356.
[28] C. Donnat and S. Holmes, "Tracking network dynamics: A survey using graph distances," *Ann. Appl. Statist.*, vol. 12, no. 2, pp. 971–1012, 2018.
[29] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos, "Network similarity via multiple social theories," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, 2013, pp. 1439–1440.
[30] K. Das, S. Samanta, and M. Pal, "Study on centrality measures in social networks: A survey," *Social Netw. Anal. Mining*, vol. 8, no. 1, pp. 1–11, 2018.
[31] R. C. Wilson and P. Zhu, "A study of graph spectra for comparing graphs and trees," *Pattern Recognit.*, vol. 41, no. 9, pp. 2833–2841, 2008.
[32] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1320–1329.
[33] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, 2018.
[34] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, and W. Wang, "SimGNN: A neural network approach to fast graph similarity computation," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 384–392.
[35] N. M. Kriege, F. D. Johansson, and C. Morris, "A survey on graph kernels," *Appl. Netw. Sci.*, vol. 5, no. 1, pp. 1–42, 2020.
[36] M. Bolla, "Spectra, Euclidean representations and clusterings of hypergraphs," *Discrete Math.*, vol. 117, no. 1–3, pp. 19–39, 1993.
[37] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1601–1608.
[38] O. Kardos, A. London, and T. Vinkó, "Stability of network centrality measures: A numerical study," *Social Netw. Anal. Mining*, vol. 10, no. 1, pp. 1–17, 2020.
[39] S. P. Borgatti, K. M. Carley, and D. Krackhardt, "On the robustness of centrality measures under conditions of imperfect data," *Social Netw.*, vol. 28, no. 2, pp. 124–136, 2006.
[40] A. R. Benson, "Three hypergraph eigenvector centralities," *SIAM J. Math. Data Sci.*, vol. 1, no. 2, pp. 293–312, 2019.
[41] 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.
[42] G. Li, L. Qi, and G. Yu, "The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory," *Numer. Linear Algebra Appl.*, vol. 20, no. 6, pp. 1001–1029, 2013.
[43] J. Chang, Y. Chen, L. Qi, and H. Yan, "Hypergraph clustering using a new Laplacian tensor with applications in image processing," *SIAM J. Imag. Sci.*, vol. 13, no. 3, pp. 1157–1178, 2020.
[44] J. Xie and L. Qi, "Spectral directed hypergraph theory via tensors," *Linear Multilinear Algebra*, vol. 64, no. 4, pp. 780–794, 2016.
[45] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[46] C. Chen, A. Surana, A. Bloch, and I. Rajapakse, "Multilinear time invariant system theory," in *Proc. Conf. Control Appl.*, 2019, pp. 118–125.
[47] C. Chen, A. Surana, A. M. Bloch, and I. Rajapakse, "Multilinear control systems theory," *SIAM J. Control Optim.*, vol. 59, no. 1, pp. 749–776, 2021.
[48] S. Ragnarsson and C. Van Loan, "Block tensor unfoldings," *SIAM J. Matrix Anal. Appl.*, vol. 33, pp. 149–169, 2012.
[49] 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.
[50] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. A1027–A1052, 2012.
[51] L. Qi, "Eigenvalues of a real supersymmetric tensor," *J. Symbolic Comput.*, vol. 40, no. 6, pp. 1302–1324, 2005.
[52] L.-H. Lim, "Singular values and eigenvalues of tensors: A variational approach," in *Proc. 1st IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, 2005, pp. 129–132.
[53] L. Chen, L. Han, and L. Zhou, "Computing tensor eigenvalues via homotopy methods," *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 1, pp. 290–319, 2016.
[54] L. Chen, L. Han, T.-Y. Li, and L. Zhou, "TenEig," 2015. [Online]. Available: https://users.math.msu.edu/users/chenlipi/TenEig.html
[55] K.-C. Chang, K. Pearson, and T. Zhang, "Perron-Frobenius theorem for nonnegative tensors," *Commun. Math. Sci.*, vol. 6, no. 2, pp. 507–520, 2008.
[56] F. Tudisco and D. J. Higham, "Node and edge eigenvector centrality for hypergraphs," *Commun. Phys.*, vol. 4, 2021, Art. no. 201.
[57] A. Gautier, F. Tudisco, and M. Hein, "A unifying Perron–Frobenius theorem for nonnegative tensors via multihomogeneous maps," *SIAM J. Matrix Anal. Appl.*, vol. 40, no. 3, pp. 1206–1231, 2019.
[58] J. Payne, "Deep hyperedges: A framework for transductive and inductive learning on hypergraphs," in *Proc. 33rd Conf. Neural Inform. Process. Syst.: Workshop Sets Partitions*, 2019.
[59] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 01, pp. 3558–3565.
[60] J. Lugo-Martinez, D. Zeiberg, T. Gaudelet, N. Malod-Dognin, N. Przulj and P. Radivojac, "Classification in biological networks with hypergraphlet kernels," *Bioinformatics*, vol. 37, no. 7, pp. 1000–1007, 2021.
[61] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognit.*, vol. 110, Art. no. 107637, 2021.
[62] B. Bollobás and B. Béla, *Random Graphs*, vol. 73. Cambridge, U.K.: Cambridge Univ. Press, 2001.
[63] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[64] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[65] B. Jhun, M. Jo, and B. Kahng, "Simplicial SIS model in scale-free uniform hypergraph," *J. Stat. Mech.: Theory Exp.*, vol. 2019, no. 12, 2019, Art. no. 123207.
[66] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
[67] G. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Proc. Neural Inf. Process. Syst.*, 2002, vol. 15, pp. 833–840.
[68] P. I. Good, *Permutation, Parametric, and Bootstrap Tests of Hypotheses*. Berlin, Germany: Springer, 2006.
[69] J. Wang and N. Zheng, "Measures of correlation for multiple variables," 2014, *arXiv:1401.4827*.
[70] I. Rajapakse and M. Groudine, "On emerging nuclear order," *J. Cell Biol.*, vol. 192, no. 5, pp. 711–721, 2011.
[71] H. Chen et al., "Functional organization of the human 4D nucleome," *Proc. Nat. Acad. Sci.*, vol. 112, no. 26, pp. 8002–8007, 2015.
[72] G. A. Dotson et al., "Deciphering multi-way interactions in the human genome," *Nature Commun.*, vol. 13, Art. no. 5498, 2022.
[73] S. Howorka and Z. S. Siwy, "Reading amino acids in a nanopore," *Nature Biotechnol.*, vol. 38, no. 2, pp. 159–160, 2020.
[74] C. Yang, R. Wang, S. Yao, and T. Abdelzaher, "Semi-supervised hypergraph node classification on hypergraph line expansion," in *Proc. 31st ACM Int. Conf. Inform. Knowl. Manage.*, 2022, pp. 2352–2361.
[75] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[76] I. A. Ebeid, M. Hassan, T. Wanyan, J. Roper, A. Seal, and Y. Ding, "Biomedical knowledge graph refinement and completion using graph representation learning and top-K similarity measure," in *Proc. Int. Conf. Inf.*, 2021, pp. 112–123.
[77] X. Li, H. Zhang, R. Wang, and F. Nie, "Multiview clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 330–344, Jan. 2022.
[78] C. Chen and Y.-Y. Liu, "A survey on hyperlink prediction," 2022, *arXiv:2207.02911*.

Amit Surana (Member, IEEE) received the bachelor's degree in mechanical engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 2000, and the M.S. degree in mechanical engineering and the M.A. degree in mathematics from Pennsylvania State University in 2002 and 2003, respectively, and his Ph.D. degree in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2007. He is currently a Technical Fellow with Raytheon Technologies Research Center. His research interests include dynamical systems, control theory, machine learning, and multi-agent systems with a broad range of applications in aerospace and defense domains.

Can Chen received the B.S. degree in mathematics from the University of California, Irvine, CA, USA, in 2016, and the M.S. degree in electrical and computer engineering and Ph.D. degree in applied and interdisciplinary mathematics from the University of Michigan, Ann Arbor, MI, USA, in 2020 and 2021, respectively. He is currently a Postdoctoral Research Fellow with the Channing Division of Network Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, MA, USA. His research interests include control theory, network science, dynamical systems, machine learning, and computational biology.

Indika Rajapakse (Member, IEEE) is currently an Associate Professor of computational medicine and bioinformatics, with the Medical School, and an Associate Professor of mathematics with the University of Michigan, Ann Arbor, MI, USA. He is also the Member of Smale Institute. His research interests include the interface of biology, engineering, and mathematics, dynamical systems, networks, mathematics of data, and cellular reprogramming.