Adaptive Neural Message Passing for Inductive Learning on Hypergraphs

Devanshu Arya, Deepak K. Gupta, Stevan Rudinac, Member, IEEE, and Marcel Worring, Senior Member, IEEE

Abstract—Graphs are the most ubiquitous data structures for representing relational datasets and performing inferences in them. They model, however, only pairwise relations between nodes and are not designed for encoding the higher-order relations. This drawback is mitigated by hypergraphs, in which an edge can connect an arbitrary number of nodes. Most hypergraph learning approaches convert the hypergraph structure to that of a graph and then deploy existing geometric deep learning methods. This transformation leads to information loss, and sub-optimal exploitation of the hypergraph’s expressive power. We present HyperMSG, a novel hypergraph learning framework that uses a modular two-level neural message passing strategy to accurately and efficiently propagate information within each hyperedge and across the hyperedges. HyperMSG adapts to the data and task by learning an attention weight associated with each node’s degree centrality. Such a mechanism quantifies both local and global importance of a node, capturing the structural properties of a hypergraph. HyperMSG is inductive, allowing inference on previously unseen nodes. Further, it is robust and outperforms state-of-the-art hypergraph learning methods on a wide range of tasks and datasets. Finally, we demonstrate the effectiveness of HyperMSG in learning multimodal relations through detailed experimentation on a challenging multimedia dataset.

Index Terms—Hypergraphs, representation learning, multimodal learning, geometric deep learning.

I. INTRODUCTION

MODELING the intrinsic properties of many real-world datasets, such as their structure and connections between the data points, requires relational data structures. Graphs are a popular data structure for discovering useful information in relational datasets due to their capability to combine object-level information with the underlying inter-object relations [71]. Data encountered in many real-world scenarios, however, contain relationships among objects which are not dyadic (pairwise) but triadic, tetradic or higher. Moreover, simplifying high-order relations to pairwise ones using a graph leads to information loss and impedes expressiveness. Thus, graphs fail to capture the collective flow of information across multiple objects. As an example, consider joint authors in research papers. Many papers will have more than two authors, yet using a graph will not be able to capture this as there are pairwise connections among the co-authors for this paper but also for other papers in which they have collaborated in different combinations. Graphs cannot model this difference. In such cases, the interactions among the objects can be fully modelled by including higher-order relations instead of pairwise relations only [70]. Some example domains where graphs are already shown to be insufficiently expressive are social networks [63], video surveillance [76] and neuroscience [33]. In all such domains the information at group level contributes to understanding the data and solving tasks. Group level and other higher-order relationships can be readily represented with hypergraphs [10], where a hyperedge can connect an arbitrary number of nodes as opposed to just two nodes in a graph.

Hypergraphs have already been shown to provide a flexible and natural framework to represent higher-order relations within homogeneous data [83]. Real-world datasets, however, often consist of objects with multiple modalities. Traditional graph- or hypergraph-based representations for multimodal data model relations in each modality independently, ignoring the heterogeneous relations between objects. Yet, such relations in the multimodal data can provide complementary information revealing fundamental characteristics of objects and their context. A simple pairwise representation of relations does not depict the collective flow of information required to do so. Superiority of hypergraph formulations is most evident in datasets and tasks naturally associated with rich higher-order relations. For instance, consider a social media platform which has users, tweets, hashtags, and images. Representing multimodal relations in the collective information generated when a user posts a tweet with an image containing multiple hashtags is infeasible using binary pairwise relations [46] or by considering tweets, users, and images as non-related channels. Hypergraphs efficiently represent multimodal objects with higher-order relations inside a channel and among channels in various domains such as visual arts [5], discussion forums [6], visual question answering systems [40], music recommender systems [14] and protein-protein interaction networks [44].

Apart from being multimodal, in real-world datasets the objects and relations among them are dynamically evolving. That
is, during training time there will be nodes which are unseen and relations which are partially observed. Such instances often exist in scenarios such as finding the most promising target audience for a marketing campaign or making movie recommendations with new movies appearing all the time. Performing inference on unseen nodes is challenging and requires an inductive learning framework.

Making inferences in relational datasets represented on a graph or hypergraph requires means to accurately propagate information across the nodes. Recent advances in geometric deep learning [12] resulted in several techniques to do so by using spectral or spatial convolution on graphs. Spectral graph convolution is based on spectral graph theory [13] and provides a well-founded mathematical framework for designing translation-invariant operators (filters). It requires, however, computing eigenvectors of large matrices, which is computationally expensive. Another shortcoming is that these filters are not localized i.e., to compute the output of a node, the convolution operation does not consider a limited number of neighborhood hops. Hence, its complexity grows with the size of the graph and therefore the method is not scalable to large datasets. This led to works applying approximation to the spectral graph convolution using $K$-localized filters which use a $K$th-order polynomial, like in ChebNet [20] and GCN [43]. Such filter localization techniques can be interpreted as spatial methods that perform message passing on local neighborhoods which are a certain number of steps away from a node. Due to their efficiency, spatial graph convolution approaches have received most attention in recent years. The resulting neural network model performs message passing operations informed by the structure of the graph to distribute encoded feature information among connected nodes [31], [42]. Message passing has proven to be effective for graph inference, but has yet to be researched for hypergraphs.

Several approaches for devising message passing in hypergraphs have been proposed aiming to exploit their expressive power [25], [74], [75]. A common implicit assumption they make is that a hypergraph is a specific type of ordinary graph. If the assumption holds, reducing the hypergraph learning problem to that of a graph should suffice. Strategies to reduce a hypergraph to a graph include transforming the hyperedges into multiple edges using clique expansion [25], [38], [80], converting to a heterogeneous graph using star expansion [1], and replacing every hyperedge with a weighted edge created using a certain predefined metric on the functional properties of the node [75]. By using a graph as an intermediate (proxy) representation, these approaches make existing graph-based message passing methods [31], [71] applicable to hypergraphs, as can be seen in the left part of Fig. 1. However, a hypergraph is not a special case of a graph. The opposite is true, graphs are simply a specific type of hypergraph [10], where the hyperedges are a superset of the pairwise edges. The complex relations in a hypergraph cannot be viewed as an instantiation of an ordinary graph, thus reducing the hypergraph problem to that of a graph cannot fully utilize the available information [22]. To address tasks in datasets with higher-order relations, a truly hypergraph-based message passing formulation is needed that ensures information propagation within and across hyperedges.

Another major limitation of the existing hypergraph representation learning frameworks is their inherently transductive nature [8], [75], thus they are inapplicable for making any inferences on unseen nodes. Often, the injection of new unseen nodes can lead to the introduction of noisy features in the message passing framework [27]. We introduce a probabilistic neighborhood sampling approach which acts as a regularizer and facilitates the proposed inductive learning framework. In a hypergraph, message passing operation on a node is different from regular graphs. This is because in a hypergraph the message propagation from the neighborhood of a node is performed at two levels - within a hyperedge and across hyperedges. Thus unlike regular non-weighted graphs, in a hypergraph not all neighborhood nodes hold equal importance. To quantify the role of a node, we identify that the uniqueness of a hypergraph lies in the detailed structure of its hyperedges as well as the distribution of nodes across them. In a graph, each edge is only shared between two nodes. A hyperedge, however, can
contain a large set of nodes, and each node could have different contributions within the hyperedge based on its relations with neighboring nodes. This is often defined by empirical measures, such as the degree centrality \cite{9,39}. Degree centrality captures how popular or active a node is in a hypergraph. Since such a paradigm is sensitive to the choice of dataset as well as the task, choosing an empirical measure is not optimal. We introduce a deterministic neighborhood attention mechanism which quantifies the importance of a node in terms of its neighborhood and the number of hyperedges to which it belongs. The proposed attention mechanism is self-adaptive to the choice of task, dataset and to any variation in the hypergraph structure. Combining such an attention mechanism with our message passing framework, we propose an inductive learning framework that exploits the full structure of hypergraphs, without any hypergraph-to-graph conversion, to perform several tasks such as node classification, link prediction, and hypergraph classification.

The points below highlight the contributions of this paper.

- We present HyperMSG, a hypergraph learning framework with a two-level message passing scheme that jointly captures the relations within a hyperedge and across the hyperedges.
- HyperMSG is inductive in nature, and facilitates probabilistic sampling of both seen and unseen nodes, based on their importance in message passing.
- HyperMSG adapts to the dataset and task by implicitly learning the importance of neighborhood nodes in representation learning.
- We demonstrate that HyperMSG outperforms the state-of-the-art methods by remarkable margins on standard benchmarks consisting of citation networks. In addition, we show that HyperMSG is highly efficient in exploring the complex heterogeneous interactions in multimodal hypergraphs to perform tasks such as multi-label classification, link prediction and recommendation. The robustness and general applicability of HyperMSG is further validated on the task of hypergraph (brain) classification in an extremely noisy neuroimaging dataset.

II. RELATED WORK

Message passing on hypergraphs aims at learning low-dimensional representations for signals (features) defined on the nodes. Recently, several methods have been proposed for message passing on graphs, and deployed for modeling physical systems, learning molecular fingerprints, predicting protein interfaces, and classifying diseases \cite{84}. However, there is a lack of an accurate message passing framework for hypergraphs. The biggest challenge in hypergraph-based learning is posed by the high variation in hyperedge cardinality i.e., the number of nodes in each hyperedge, which limits the accurate and efficient information propagation from one node to another along the hyperedges. For better understanding of message passing neural networks, we first provide a brief overview of some popular graph-based message passing neural networks. We will then discuss the existing techniques for representation learning on hypergraphs that are related to our proposed framework HyperMSG.

Message Passing in Graphs: The general idea behind a graph convolutional neural network is to define convolution on graphs, where the input is a graph instead of e.g., a 2-D image grid. Methods for performing convolution on graphs can be broadly classified into spatial (message passing) or spectral methods \cite{12}. As mentioned in the introduction, the major drawbacks of spectral approaches is their rigidity in generalizing to new graph structures and their high computational complexity \cite{82,85}. In this work, we focus on spatial approaches that use message passing neural networks. The basic framework of any message passing network takes in a graph with signals (i.e., features) on each of its nodes as input and learns embeddings for each node by aggregating information from its neighbors \cite{73}. Message passing neural networks outline a general message passing framework for learning such an aggregation mechanism on graphs. It passes information (messages) from one node to another along edges and repeats it in \( K \)-steps to let information propagate through the graph. Several variants of this general approach have been proposed, such as \cite{31,32,34,43,49}. Recently, it has been further extended to heterogeneous graphs in which nodes (and edges) are typed, allowing graphs to incorporate auxiliary information. Some of these ideas include using attention-based mechanism such as typed attention \cite{52}, neighbour attention \cite{79}, vertex-level and semantic-level attention \cite{67} across different types of nodes or using metapath based aggregations \cite{29}. Unlike graph-based models with only binary relations, hypergraph learning models need to explore the higher-order relations in the data \cite{78}.

Message Passing in Hypergraphs: Since the introduction of learning with hypergraphs \cite{83}, several such methods have been introduced \cite{30}, and successfully deployed in various tasks, such as link prediction \cite{46}, community detection \cite{16} and visual object tracking \cite{69}. In spectral theory of hypergraphs, methods have been proposed that fully exploit the hypergraph structure using non-linear Laplacian operators \cite{15,35}. However, these methods have similar drawbacks as spectral graph methods with regard to their computational complexity and scalability. Learning on the hypergraph can also be seen as the process of message passing along the hypergraph structure in analyzing the structured data. Emulating a graph-based message passing framework for hypergraphs is not straightforward since a hyperedge involves more than two nodes which makes the interactions inside each hyperedge complex. Hypergraphs are mathematically represented using either an incidence matrix or an adjacency tensor. Incidence matrix based representations of hypergraphs are rigid in describing the structures of higher order relations \cite{48}. On the other hand, formulating message passing on a higher-dimensional representation of a hypergraph using adjacency tensors makes it computationally expensive and restricted to small datasets and uniform hypergraphs \cite{81}.

To circumvent the above issues, \cite{25} and \cite{8} reduce a hypergraph to a graph using clique expansion and perform graph convolutions on them. Further, \cite{38} assume the initial hypergraph structure is weak, and extend the work of \cite{25} to construct dynamic hypergraphs. Recently proposed HyperGCN replaces a hyperedge with pair-wise weighted edges between vertices called mediators \cite{75}. The weights are calculated by comparing...
the functional properties of neighboring nodes (e.g., using \( \text{arg-max} \) over them) and hence lose the structural information within a hypergraph. With the use of mediators, HyperGCN can be interpreted as an improved variant of clique expansion, and to the best of our knowledge, is also the state-of-the-art method for all the hypergraph representation learning methods, where a graph-based message passing neural network is eventually used. However, it still suffers from the same limitation as the clique expansion [22]. These limitations are further discussed on the example of a Fano plane in the Appendix, available online. Furthermore, these approaches are inherently transductive and thus, as indicated earlier, cannot perform inference on unseen nodes. In [58] a generalized hypergraph learning framework is presented that uses random walks, however its message passing framework is not robust for heterogeneous hypergraphs and its validity for inductive setting is still unexplored. None of the above approaches utilize the complete structural information in the hypergraph, leading to sub-optimal learning performance.

In our preliminary work we introduced HyperSAGE [3], an inductive representation learning framework for hypergraphs that can exploit their full structure by aggregating messages in a two-stage procedure. HyperSAGE introduced a generalized message passing approach on hypergraphs by taking motivation from graph-based message passing neural networks. HyperSAGE was an adaptation of the GraphSAGE framework for hypergraphs. GraphSAGE, one of the most popular graph neural network (GNN) models, generates embeddings by sampling and aggregating features from a node’s local neighborhood. In generalizing and adapting GraphSAGE for hypergraphs, HyperSAGE extends this concept to handle the multi-node connections inherent in hyperedges. The core idea is to aggregate features from nodes within the same hyperedge to generate node embeddings. This approach allows HyperSAGE to learn from the complex relationships in hypergraphs, but it essentially treats each node in a hyperedge equally, thereby failing to capture the importance of the node in the overall context of the hypergraph data structure. It managed to achieve near state-of-the-art performance on benchmark datasets. However, it suffers from lack of adaptability to datasets and structural variation in hypergraphs and poor parallelism. Huang et al. [36] further built on our proposed HyperSAGE framework to circumvent these issues and generalize the aggregation approach. Recently, [21] proposed message passing in hypergraphs using a dual attention mechanism for classifying text in documents by constructing multiple types of hyperedges containing node \( v \) or all nodes in \( \mathcal{E} \), or all nodes in \( \mathcal{E} \). The proposed message passing framework combines both types of neighborhood information (within hyperedges and across hyperedges) separately and thus mitigates the shortfall of hypergraph to graph conversion. However, the proposed dual attention mechanism uses the functional properties (i.e., node features) of the hypergraph. This leads to insufficient information propagation as it fails to quantify the importance of a node with respect to its neighbors within and across hyperedges, which is an important aspect for capturing the structural properties of a hypergraph. In this work, we propose HyperMSG, which eliminates matrix- or tensor-based formulations in its neural message passing scheme for hypergraphs. It is further inductive and utilizes all the available information in a hypergraph.

### III. Hypergraph Preliminaries

In this section, we first introduce some of the preliminary definitions and notations of hypergraphs which will be used to formulate the HyperMSG framework.

**Definition 1 (Hypergraph):** A hypergraph \( \mathcal{H} \) is represented as \( \mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X}) \), where \( \mathcal{V} = \{v_1, v_2, \ldots, v_N\} \) denotes a set of \( N \) nodes and \( \mathcal{E} = \{e_1, e_2, \ldots, e_K\} \) a set of hyperedges, with each hyperedge comprising a non-empty subset from \( \mathcal{V} \). \( \mathbf{X} \in \mathbb{R}^{N \times d} \) denotes the feature matrix, with the feature vector \( x_i \) corresponding to the respective \( v_i \) column.

The cardinality of any hyperedge \( e_i \) is the number of nodes contained in that hyperedge, given by \( |e_i| \). Unlike in a graph, the hyperedges of \( \mathcal{H} \) can contain different number of nodes i.e., \( 1 \leq |e_i| \leq |\mathcal{V}| \). Definition 1 makes it clear that graphs are simply a special case of hypergraphs with a fixed cardinality of 2 for all the edges.

We define three different types of neighborhoods as follows.

**Definition 2 (Intra-Edge Neighborhood):** The intra-edge or local neighborhood of a node \( v_i \in \mathcal{V} \) for any hyperedge \( e \in \mathcal{E} \) is defined as the set of nodes \( v_j \), belonging to \( e \), excluding node \( v_i \) and is denoted by \( \mathcal{N}(v_i, e) \).

The intra-edge neighborhood of a node captures the higher order relationships and provides localized group level information to it. Further, let \( E(v_i) = \{e \in \mathcal{E} | v_i \in e\} \) be the set of hyperedges containing node \( v_i \). The degree of node \( v_i \) is thus given by \( |E(v_i)| \).

**Definition 3 (Inter-Edge Neighborhood):** The inter-edge or global neighborhood of a node \( v_i \in \mathcal{V} \), is defined as the neighborhood of \( v_i \) spanning across the set of hyperedges \( E(v_i) \) and is given by \( \mathcal{N}(v_i) = \bigcup_{e \in E(v_i)} \mathcal{N}(v_i, e) \).

The inter-edge or global neighborhood of a node captures its global positioning and gathers information from hyperedges similar to the node. Finally, the condensed neighborhood parameterized by \( \alpha \) gives a subset of nodes within a hyperedge.

**Definition 4 (Condensed Neighborhood):** The condensed neighborhood of any node \( v_i \in \mathcal{V} \) is defined as the sampled set \( \mathcal{N}(v_i, e; \alpha) \) comprising of \( \alpha \) nodes from a hyperedge \( e \in E(v_i) \), if \( \alpha < |e| \), or all nodes in \( e \) if \( \alpha = |e| \).

### IV. Proposed Model

The main concept behind devising a message passing neural network on hypergraphs is to aggregate feature information from the neighborhood of a node which spans across multiple hyperedges with varying cardinality. In this section we propose HyperMSG, a framework that performs message passing at two levels for a hypergraph. Further, we discuss our adaptive framework that implicitly learns the importance of each node in the representation learning process. Our approach inherently allows inductive learning, which makes it also applicable on hypergraphs with unseen nodes.

#### A. Two-Level Message Passing Framework

We propose to interpret the propagation of information in a given hypergraph as a two-level aggregation problem, where the neighborhood of any node is divided into its *intra-edge* neighbors and *inter-edge* neighbors. This information is present
in the form of signals on each node often referred to as message as mentioned in the previous section. For message aggregation, we define the aggregation function as a permutation invariant set function on a hypergraph \( \mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X}) \) that takes as input a countable unordered message set and outputs a reduced or aggregated message of the same dimension as the original message. Further, for two-level aggregation, let \( \mathcal{F}_1(\cdot) \) and \( \mathcal{F}_2(\cdot) \) denote the intra-edge and inter-edge aggregation functions, respectively. Schematic representation of the two functions is provided in Fig. 2. Similar to \( \mathbf{X} \), we also define \( \mathbf{Z} \in \mathbb{R}^{N \times l} \) as the aggregated feature matrix built using the outputs \( \mathbf{z}_i \) with dimension \( l \) from the aggregation functions. Message passing at node \( v_i \) can then be expressed as

\[
\begin{align*}
  s_1 & \leftarrow \{ \mathbf{x}_j \mid v_j \in \mathcal{N}(v_i; e; \alpha) \}, \\
  s_2 & \leftarrow \{ \mathcal{F}_1^\ast (s_1) \mid e \in \mathcal{E}(v_i) \}, \\
  \mathbf{z}_i & \leftarrow \mathbf{x}_i \oplus \mathcal{F}_2(s_2),
\end{align*}
\]

where \( s_1 \) and \( s_2 \) denote the unordered sets of feature vectors and intra-edge aggregations, respectively.

To ensure that the expressive power of a hypergraph is preserved or at least the information loss is minimized, the choice of aggregation function should comply with certain properties. First, it should capture the features of neighborhood nodes in a manner that is invariant to permutation of nodes and hyperedges within its neighborhood. This ensures the final message propagation to a node does not depend on the sequence in which we pick up its neighbors. Many graph-based methods use aggregation functions, such as mean and max functions [43]. Mean and max-pooling aggregators are well-defined multiset functions because they are permutation invariant and these functions learn different attributes of the neighborhood (max learns distinct elements and mean learns distributions). These aggregations have proven to be successful for node classification problems [34], [43]. However, they are not injective and hence have their limitations in learning unique representations in various cases such as when the nodes have repeating features [73] or when the features are continuous [18]. There are other non-standard neighbor aggregation schemes that we do not cover, e.g., weighted average via attention [66], LSTM pooling [34], [55] and stochastic aggregations [68]. We emphasize that our theoretical framework is general enough to characterize the representational power using a family of generalized mean aggregation functions. In the future, it would be interesting to apply our framework to analyze and understand other aggregation schemes.

Property 1. (Hypergraph Isomorphic Equivariance): A message aggregation function \( \mathcal{F}(\cdot) \) is equivariant to hypergraph isomorphism, if for two isomorphic hypergraphs \( \mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X}) \), \( \mathcal{H}' = (\mathcal{V}', \mathcal{E}', \mathbf{X}') \), and permutation operator \( \sigma \), given that \( \mathcal{H}' = \sigma \cdot \mathcal{H} \) and \( \mathbf{Z} \) and \( \mathbf{Z}' \) represent the aggregated feature matrices obtained using \( \mathcal{F}(\cdot) \) on \( \mathcal{H} \) and \( \mathcal{H}' \) respectively, the condition \( \mathbf{Z}' = \sigma \cdot \mathbf{Z} \) holds.

Second, the aggregation function should also preserve the global neighborhood invariance at the ‘dominant nodes’ of the graph. Here, dominant nodes refer to nodes that act as hubs in power-law networks, possessing many more connections than their neighbors [2]. The aggregation function should ideally be insensitive to whether the provided hypergraph contains a few large hyperedges, or a large number of smaller ones obtained from splitting them. Generally, a hyperedge would be split in a manner that the dominant nodes are shared across the resulting hyperedges. In such cases, global neighborhood invariance would imply that the aggregated output at these nodes before and after the splitting of any associated hyperedge stays the same. Otherwise, the learned representation of a node will change significantly with each split. Based on these considerations, we define the following properties for a generic message aggregation function.

Property 2. (Global Neighborhood Invariance): A message aggregation function \( \mathcal{F}(\cdot) \) satisfies global neighborhood invariance at any node \( v_i \in \mathcal{V} \) for a given hypergraph \( \mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X}) \), if for any hyperedge \( e \in \mathcal{E}(v_i) \) being split into multiple hyperedges without changing \( \mathcal{N}(v_i) \), and \( \mathbf{z}_i \) and \( \mathbf{z}_i' \) denoting the aggregated feature vectors obtained before and after splitting, the condition \( \mathbf{z}_i' = \mathbf{z}_i \) holds.

Aggregation Using Generalized Means: One major advantage of our strategy is that the message passing module is decoupled from the choice of the aggregation method itself. This allows our approach to be used with a broad set of generalized means aggregation functions, a powerful and rich family of aggregation functions that have been shown to perform well for graph representation learning [47]. The permutation invariant nature of generalized means makes them satisfy Property 1. Further, we show that with appropriate combinations of the intra-edge and inter-edge aggregations, Property 2 is also satisfied.

Mathematically, generalized means can be expressed as \( M_p = (\frac{1}{p} \sum_{i=1}^{n} x_i^p)^{\frac{1}{p}} \), where \( n \) refers to the number of elements to aggregate, and \( p \) denotes its power. The choice of \( p \) allows providing different interpretations to the aggregation function. For example, \( p = 1 \) denotes arithmetic mean aggregation, \( p = 2 \) refers to mean squared estimate and a large value of \( p \) approximates max pooling from the group. Similarly, \( M_p \) can be used for approximating geometric means with \( p \to 0 \). We use generalized means for intra-edge as well as inter-edge aggregation. The two functions \( \mathcal{F}_1(\cdot) \) and \( \mathcal{F}_2(\cdot) \) as stated in Section IV-A,
for aggregation at node $v_i$ are then defined as
\[
\mathcal{F}_1(s_1) = \left( \frac{1}{|\mathcal{N}(v_i)|} \sum_{v_j \in \mathcal{N}(v_i, e)} w_j x_j^p \right)^{1/p},
\]
\[
\mathcal{F}_2(s_2) = \left( \frac{1}{|E(v_i)|} \sum_{e \in E(v_i), s \in s_2} s^p \right)^{1/p}.
\]

For Property 2 to hold in (4) and (5) above, power term $p$ needs to be the same for $\mathcal{F}_1$ and $\mathcal{F}_2$. Also, the scaling term $w_j$ needs to be added to balance the bias in the weighting introduced in intra-edge aggregation due to varying cardinality across the hyperedges. The related mathematical proof is presented in the Appendix, available online.

B. Learning the Importance of Nodes

Commonly, the contributions from the neighboring nodes within a hyperedge are weighted equally through the choice of simple aggregation functions such as $\text{mean}$ and $\text{max}$, among others. Thus, the importance of any node is defined by only its functional characteristics - the values that are contained in its feature vector. However, hypergraphs also possess a complex structure, and this component is almost unused in the learning process.

We hypothesize that analyzing the structural information can reveal the importance of each node and enhance the message passing process. The structural information of a node $v$ in a hypergraph can be primarily defined by two terms: its global neighbourhood set $\mathcal{N}(v)$ and its hyperedge set $E(v)$. To quantify this information for a node in any graph, [28] introduced a degree centrality measure which is simply the total number of edges incident on that node. However, compared to graphs where $|\mathcal{N}(v)| = |E(v)|$, these two terms can be significantly different for a hypergraph depending on its structure. Defining empirical degree centrality functions to measure the importance of any node in hypergraphs was shown to be effective, however, such functions are sensitive to the choice of dataset and task [39].

We introduce a learnable function to quantify the importance of a node which is expressed in terms of $\mathcal{N}$ and $E$ as $\mathcal{F}(\mathcal{N}, E)$, a Multi-Layer Perceptron (MLP) which can be mathematically represented as: $\mathcal{F}(\mathcal{N}, E) = f(W_2 \cdot g(W_1 \cdot |\mathcal{N}(v)| E(v)) + b_1) + b_2$. Here, $(W_1)$ and $(W_2)$ are the learnable weight matrices for the first and second layers, respectively, while $b_1$ and $b_2$ are the bias terms. $g(\cdot)$ represents the activation function for the hidden layer, and $f(\cdot)$ is the activation function for the output layer. For hypergraphs, node importance function $\mathcal{F}(\cdot)$ is a complex non-linear mapping. Thus, rather than empirically choosing a single function, we learn the importance value for each node using a small fully-connected neural network comprising two hidden layers. For any node $v$, the output of $\mathcal{F}(\cdot, \cdot)$, denoted by $C$, can be interpreted as an attention weight that defines the importance of that node in the message passing process (see Fig. 2). Let $C_j$ denote the attention weight for node $v_j \in \mathcal{N}(v_i, e)$. We conjecture that using $C$ as a weighting term for each node during message passing can improve the learned node embeddings in a hypergraph irrespective of dataset or task.

C. Inductive Learning on Hypergraphs

Inductive learning of nodes is a challenging problem for hypergraphs as it requires the model to generalize on previously unseen nodes with a diverse set of features and associated sub-hypergraphs. Most existing approaches are inherently transductive as they make predictions on nodes in a single, fixed hypergraph. These approaches directly optimize the node representations using matrix-factorization-based objectives, and thus do not generalize to unseen data. HyperMSG tackles this challenge through the use of a neural message passing framework as described in Section IV-A that learns to generate embeddings by sampling and aggregating features from a node’s local neighborhood. Our approach uses a neural network comprising $L$ layers, and feature-aggregation is performed at each of these layers, as well as across the hyperedges. Algorithm 1 describes the forward propagation mechanism, which implements the aggregation function introduced above. At each iteration, nodes first aggregate information from their neighbors within a specific hyperedge. This is repeated over all the hyperedges across all the $L$ layers. The trainable weight matrices $W^l$ with $l \in L$ are used to aggregate information across the feature dimension and propagate it through the hypergraph. The representation on any unseen node can then be obtained by an aggregation process similar to that on seen nodes.

**Algorithm 1:** HyperMSG Inductive Message Passing.

**Input:** $\mathcal{H} = (V, E, \mathcal{X})$; depth $L$; weight matrices $W^l$ for $l = 1 \ldots L$; non-linearity $\sigma$; intra-edge aggregation function $\mathcal{F}_1(\cdot)$; inter-edge aggregation function $\mathcal{F}_2(\cdot)$; node importance function $C(\cdot, \cdot)$

**Output:** Node embeddings $z_i | v_i \in \mathcal{V}$

\[
h_0^l \leftarrow x_i \in \mathcal{X} | v_i \in \mathcal{V}
\]

for $l = 1 \ldots L$

\[
h_i^l \leftarrow h_i^{l-1} \cdot \mathcal{F}_1(s_1)
\]

end

\[
h_i^l \leftarrow \mathcal{F}_2(s_2)
\]

end

$z_i \leftarrow h_i^L | v_i \in \mathcal{V}$

**Probabilistic Sampling-Based Aggregation:** The modular framework of HyperMSG provides flexibility in adapting the message aggregation module to fit a desired computational memory. This is achieved through aggregating information from only a condensed neighborhood set (Definition 4) $\mathcal{N}(v_i, e; \alpha)$ instead of the full neighborhood $\mathcal{N}(v_i, e)$. We propose to apply
sub-sampling only on the nodes from the training set, and use information from the full neighborhood for the test set. The advantages of this are twofold. First, a reduced number of samples per aggregation at training time reduces the memory capacity requirement. Second, similar to dropout [62], it serves to add regularization to the optimization process. Using the full neighborhood on test data avoids randomness in the test predictions, and generates consistent output.

For the sampling process, we choose to perform probabilistic selection of the nodes for aggregation based on their importance in the hypergraph using the learned attention weights as described in Section IV-B. The probability $P_j$ that the node $v_j$ gets selected in the sampled subset at any iteration is then given by

$$P_j = \frac{C_j}{\sum_{j=0}^{|\mathcal{N}(v_i)|} C_j}$$

(6)

implying that the neighbor node judged as more important for $v_i$ should be sampled more often.

**Multimodal Hypergraph Construction:** Multimodality in hypergraph context refers to hyperedges that connect nodes across different types or modes of data, presenting unique challenges in terms of representation and learning. Heterogeneous hypergraphs involve nodes and hyperedges of different types, focusing on the diversity of elements within the graph structure. Multimodal hypergraphs, on the other hand, model multiple content modalities (e.g., visual, audio and text), as well as their complex interactions. To construct a multimodal hypergraph within the HyperMSG framework, we adopt a systematic methodology that ensures the comprehensive integration of various data modalities and their intricate interconnections. Initially, nodes representing different data types are identified and categorized based on their respective data modalities, such as textual, visual, or relational data. Hyperedges are crafted to represent not just binary relationships, but also complex interactions that involve multiple nodes across different modalities, effectively capturing the essence of multimodal data. This includes, for example, the association between a textual description and an image or the relationship between a user’s social media activity and their consumption patterns. Special attention is given to ensuring that these hyperedges accurately reflect the multifaceted nature of the interactions, incorporating frequencies or attributes where necessary to highlight the significance of each hyperedge.

**Time Complexity Analysis:** The time complexity for training HyperMSG is $O(Td(1 + h(d + c)))$ where, $T$ is the total number of training iterations, $d$ denotes the dimensionality of the input feature vector, $h$ is the number of hidden layers, and $c$ is the number of classes. Thus, the time complexity of HyperMSG is at par with HGN [25] and HyperGCN [75], with an added advantage of probabilistic sampling by using $\alpha$ that can reduce the memory constraints while also acting as a regularizer. Additional details are provided in the supplementary material, available online.

**V. EXPERIMENTS**

We perform a variety of experiments to evaluate HyperMSG and compare its performance with other hypergraph based learning methods. First, the performance of HyperMSG is evaluated on the task of semi-supervised node classification in hypergraphs through several experiments on representative benchmark datasets. This experiment is performed to test the node-level representative learning capability of HyperMSG and analyze all its possible variants. The results are compared with the state-of-the-art hypergraph representation learning methods. Second, we study the stability of HyperMSG by proposing the task of hypergraph classification on an extremely noisy brain neuroimaging dataset. Finally, to show the efficiency of HyperMSG in performing multimodal learning, we compare its performance with recent hypergraph learning methods on a social multimedia dataset. We further show the advantage of using our inductive framework to perform node classification of previously unseen nodes in the multimedia dataset.

**A. Semi-Supervised Node Classification in Academic Network**

**Experimental Setup:** We use the standard co-citation and co-authorship network datasets: CiteSeer, PubMed, Cora [60], DBLP [59] and arXiv [17] for this experiment where the task is to predict the topic to which a document belongs (multi-class classification). The input feature vector $x_i$ corresponds to a bag of words, where $x_{i,j} \in x_i$ is the normalized frequency of occurrence of the $j^{th}$ word. Further, for all experiments, we use a two-layered neural network. All models are implemented in Pytorch and trained using Adam optimizer [41]. Additional implementation details are provided in the Appendix, available online.

**Significance of Learning the Importance of Nodes:** For a better understanding on the significance of learning a node importance function for hypergraphs, we show example distributions of $|\mathcal{N}(v)|/|E(v)|$ for CORA co-citation and co-authorship datasets (see Fig. 3). For the CORA co-citation dataset, $|\mathcal{N}(v)|/|E(v)|$ is mostly close to 1. For this data, we found that the performance gain with message passing using HyperMSG over the graph-based model is relatively small. On the contrary, for CORA co-authorship, the distribution of $|\mathcal{N}(v)|/|E(v)|$ is right-skewed, and has values significantly higher than 1 as well. For this data, HyperMSG outperforms graph-based models for node classification by more than 5%. This experiment illustrates that even with the same functional

![Fig. 3. Distribution of $|\mathcal{N}(v)|/|E(v)|$ values across CORA co-citation and CORA-co-authorship datasets.](image)
characteristics (features) of nodes, the structural information encoded within a hypergraph plays a key role in learning better node embeddings. Additional analysis is presented in the Appendix, available online.

Effect of Generalized Mean Aggregations: We study here the effect of different choices of $p$ in the proposed two-level aggregation function on the performance of HyperMSG. Fig. 4 shows the accuracy scores obtained for 4 different choices of $p$ and $\alpha$ on DBLP and Pubmed datasets. Across all values of $p$, we observe that $p = 1$ works best for both datasets. For other choices of $p$, the performance of the model is reduced. For $\alpha = 2$, performance of the model seems to be independent of the choice of $p$ for both the datasets. A possible explanation could be that the number of neighbors is very small, and change in $p$ does not affect the propagation of information significantly.

Effect of Probabilistic Sampling: We study here the effect of number of samples per aggregation on the performance of the model (Fig. 4). For DBLP, model performance increases with increasing value of $\alpha$. However, for Pubmed, we observe that performance improves up to $\alpha = 8$, but then a slight drop is observed for larger sets of neighbors. Note that for Pubmed, the majority of the hyperedges have cardinality less than or equal to 10. This means that for $\alpha = 16$, information will be aggregated from almost all the neighbors, thereby involving almost no random sampling. Stochastic sampling of nodes can serve as a regularization mechanism and reduce the impact of noisy hyperedges. This is possibly the reason that performance for $\alpha = 8$ is higher than for 16.

Performance Comparison With Existing Methods: In Table I, we compare the results of HyperMSG with state-of-the-art hypergraph learning methods. Among these, the homogeneous networks use either the co-citation and co-authorship datasets. The heterogeneous networks combine information from both co-citation and co-authorship datasets. For HyperMSG, we use homogeneous network for a fair comparison. We report the results with arithmetic mean $p = 1$ using the complete neighborhood i.e., $\alpha = |e|$. To show the significance of learning importance of nodes, we report the results for HyperMSG as well as its non-adaptive variant in Table I. For all models, 10 data splits over 8 random weight initializations are used, totalling 80 experiments per method for every dataset. The data splits are the same as in HyperGCN and are described in the Appendix, available online. Note that for Pubmed and Citeseer dataset, the co-authorship information does not exist and hence, the heterogeneous models are not applicable.

From Table I, we observe that both variants of HyperMSG outperform the homogeneous networks by remarkable margins. Interestingly, our implementations of HyperMSG with only homogeneous information are able to outperform the heterogeneous networks that use information from two different sources. This clearly demonstrates that HyperMSG exhibits
strong representative power and is able to extract information from the hypergraphs at levels beyond the existing baselines.

**Stability Analysis:** We further study the stability of our method in terms of the variance observed in performance for different train-test split ratios. We compare the results with HyperGCN under similar settings, as it is the state-of-the-art method of a broad set of hypergraph learning methods which are based on some sort of hypergraph to graph conversion. Fig. 5 shows results for the two learning methods on 5 different train-test ratios. We see that the performance of both models improves when a higher fraction of data is used for training, and the difference in their performances decreases at the train-test ratio of 1/3. However, for smaller ratios, we see that HyperMSG outperforms HyperGCN across all datasets. Further, the standard deviation for the predictions of HyperMSG is lower than that of HyperGCN. Clearly, this implies that HyperMSG is able to better exploit the information contained in the hypergraph compared to HyperGCN, and can thus produce more accurate and stable predictions. Results of the experiment on Cora and Citeseer, which exhibit a similar trend, can be found in the Appendix, available online.

**Classification of Unseen Nodes:** To test the performance of HyperMSG on unseen nodes, we create inductive learning datasets for DBLP, Pubmed and Citeseer. To do so, we split each dataset into a ratio of 1:3:1 for the training, validation (seen) and test (unseen) sets, respectively. To obtain the unseen test set, we break the hypergraph into two sub-hypergraphs. Note that this splitting leads to several node connections being disregarded as well as a relatively sparse test hypergraph. This can induce noise in the learning process, which we tackle by employing our probabilistic sampling mechanism in message passing. We show that by capturing all the structural and functional properties of a hypergraph, HyperMSG performs better than the other models. Table II presents the performance scores of HyperMSG for different choices of \( \alpha \). To the best of our knowledge, no competitive baseline inductive learning method existed before our preliminary work HyperSAGE [3] which introduced a two-level message aggregation framework. Recently, [36] further extended the two-level message aggregation framework and proposed a set of inductive learning methods. In this experiment, we compare the performance of HyperMSG with the base MLP+HLR approach and UniGCN [36] as a reference. A general observation is that HyperMSG works well for unseen nodes and significantly outperforms the other models except in the case of DBLP. Further, the difference between performance scores for seen and unseen nodes is stable across all datasets. Based on these observations, it can be concluded that HyperMSG works well in an inductive setting, i.e., on unseen nodes, as well.

### B. Hypergraph Classification on Neuroimaging Data

To demonstrate the generality of HyperMSG, we use it on the task of hypergraph classification. For performing hypergraph classification, the model needs to integrate the learned features from each node in a way that the combined features are significantly dissimilar across hypergraphs. This makes hypergraph classification a challenging task as compared to the previous node classification tasks. Further, to also study the robustness of the proposed model under noisy scenarios, we choose an extremely noisy brain neuroimaging dataset [19], where each brain image is a hypergraph. It involves classification of control subjects for autism spectrum disorder (ASD) using 4D resting-state functional magnetic resonance imaging (fMRI) data. Typically, an fMRI sample comprises about 20,000 voxels with 300 time points, making it extremely high-dimensional with a significant level of inherent noise [24], [56].

Some earlier methods have sought to mitigate this problem through aggregating information along one of the dimensions, thus leading to a 3D volume [23], [65]. Other approaches involve summarization of the data as cross-correlation matrix between macro regions of the brain [4], [57]. Although these solutions simplify data handling, they come at the expense of significant information loss. Hypergraphs can be used to handle this task without any such approximations. An example representation of a brain hypergraph is shown in Fig. 6. More details about the processing steps and data preparation are provided in the Appendix, available online.
Table III shows the performance scores for the various hypergraph-based methods. Due to class imbalance in the dataset, we use AUC-ROC to measure the performance. We see that both variants of HyperMSG outperform HGNN, HyperGCN and UniGCN methods, with our adaptive variant showing an improvement of around 5% over HGNN. The results show the robustness and stability of HyperMSG in utilizing the information within hypergraphs. The large variance observed in performance for all the methods can be attributed to the low signal-to-noise ratio of the dataset.

C. Multimodal Hypergraph Analysis

Social networks are rich with multimodal information and learning an effective representation for the entities of interest in them, such as users, images and text, has gained a great attention in different applications. Examples include node classification [11], [53], link prediction [46], [51], community detection [6], [45], and network visualization [26], [64]. In this section, we evaluate the performance of HyperMSG on MIR Flickr [37], a social multimedia dataset commonly used for multimodal learning on hypergraphs [5], [7], [72].

Experimental Setup: The MIR Flickr dataset contains heterogeneous entities and relationships among them. In particular, the dataset consists of 25,000 images (I) from Flickr posted by 6,386 users (U) to 10,575 groups (G) and annotated with over 50,000 user-provided tags (T). The dataset also provides manually-created ground truth image annotations at the semantic concept level. There are 25 such unique concepts in the dataset. Similar to the related multimedia works [5], [50], we remove noisy tags occurring less than 50 times and obtain a vocabulary consisting of 457 tags. Accordingly, 18 concepts are preserved and utilized to validate the performance in our experiments. In this work, we follow [7] as a testbed for demonstrating multimodal learning capabilities of HyperMSG by performing 3 types of tasks: Task 1: Multi-Label Image Classification, Task 2: Image-User Link Prediction and Task 3: Group Recommendation.

Hypergraph Construction: We create a hypergraph where each node represents an image and the hyperedges (relations between images) are formed using the tags, similar to our previous work [7]. For all the three tasks hyperedges correspond to the image-tag relationships i.e., a hyperedge is constructed among images sharing a common tag. Subsequently, each image is associated with ground truth labels, users and groups respectively for the corresponding three tasks. Fig. 7 shows the image-tag hypergraph (left) and the predicted classes (right) i.e., the ground-truth labels in case of Task 1, users in case of Task 2 and groups in case of Task 3.

Performance Comparison With Existing Methods: We compare the performance of our proposed HyperMSG model with the following hypergraph learning methods: HyperGCN [75], HGNN [25], HyperLearn [5] and H_{GDL} [7]. For a fair and thorough comparison, we conduct the experiments using only structural information and hence we assign identity features to each image (node of the hypergraph). In the case of Task 1, we provide a detailed performance analysis on each of the concepts. We report the performance of our proposed HyperMSG approach and the alternatives in terms of Average Precision (AP), a standard evaluation measure used for multi-label image classification benchmarks. For the other two tasks, we combine the classification scores for all the users/groups and plot their respective Receiver Operating Characteristics (ROC) curves. The ROC curve depicts how well a model is able to predict the presence/absence of a class among images with the corresponding metadata. Fig. 8 shows that both variants of HyperMSG consistently outperform the other hypergraph learning methods in Task 1, i.e., classifying images into the 18 concepts. The results of this experiment suggest that our proposed approach is more effective in modeling social image-tag relationships than the state-of-the-art alternatives. The advantage of HyperMSG in
Fig. 8. Task 1 (Multi-Label Image Classification) Detailed performance comparison in terms of Average Precision over 18 concepts on the MIRFlickr dataset.

Fig. 9. Receiver Operating Characteristics (ROC) curve showing the performance of the models on (a) Image-User Link Prediction and (b) Group Recommendation. On both tasks HyperMSG consistently outperforms alternative multimodal hypergraph learning methods.

TABLE IV
AVERAGE PRECISION OF HYPERMSG AND BASELINE HYPERGRAPH LEARNING METHODS FOR THE TASK OF MULTI-LABEL IMAGE CLASSIFICATION ON MIR FLICKR DATASET ON BOTH SEEN AND UNSEEN NODES

| Method                     | Seen       | Unseen     |
|----------------------------|------------|------------|
| MLP+HLR                    | 0.64 ± 1.3 | 0.61 ± 1.4 |
| UniGCN                     | 0.70 ± 1.1 | 0.67 ± 1.3 |
| HyperSAGE                  | 0.67 ± 1.5 | 0.61 ± 1.9 |
| HyperMSG (non-adaptive, ours) | 0.70 ± 1.1 | 0.68 ± 1.3 |
| HyperMSG (ours)            | 0.73 ± 0.9  | 0.69 ± 1.1  |

learning multimodal relations as compared to the other hypergraph methods is further confirmed by the ROC curves shown in Fig. 9 for Task 2 and Task 3. In all three experiments we observe that the adaptive version of HyperMSG utilizing attention-based mechanism comes out as the best performer. We hypothesize that this can be attributed to the significance of global and local neighborhoods in social multimedia networks where the nodes that have many connections (high-degree nodes) tend to be connected to the other nodes with many connections, while they are surrounded by many small clusters of low-degree nodes [54].

Inductive Learning: To evaluate the performance of HyperMSG on unseen nodes, we perform multi-label image classification on the MIR Flicker dataset. For this experiment, we include the features of the images as well, which we extract from the penultimate layer of a pre-trained VGG-16 network [61]. Similar to Section V-A, we compare our method with inductive learning methods, MLP+HLR and UniGCN and report the average precision (AP) scores averaged over all the classes. We split the dataset into the training, validation (seen) and test (unseen) samples according to a 1 : 3 : 1 ratio. To obtain the unseen test set, we break the hypergraph into two sub-hypergraphs, ensuring that the training set contains at least one image from each class. Table IV shows that the HyperMSG performs better than the alternatives on both seen and unseen nodes, while also producing a lower standard deviation in the performance level.

VI. CONCLUSION

In this paper, we have presented HyperMSG, a two-level neural message passing framework for inductive learning on hypergraphs. HyperMSG fully utilizes the inherent higher-order relations in a hypergraph structure without reducing it to an intermediate graph representation. It adaptively learns the importance of each node during the learning process, thereby improving the message passing process. Through experiments on several representative datasets, we have shown that HyperMSG outperforms the existing methods for hypergraph learning. We have demonstrated that HyperMSG generates stable predictions for very sparse sampling as well as when the nodes are unseen during the training process. From the results on the challenging task of brain image classification for autism, we conclude that HyperMSG yields accurate and robust results in hypergraph classification even on extremely noisy time-series datasets. Finally, HyperMSG is suitable for performing mutimodal representation learning tasks and can outperform all existing hypergraph based learning methods on information rich multimedia data.

ACKNOWLEDGMENT

The authors would like to thank Selene Gallo, Dr. Rajat Thomas and their team at Amsterdam Medical Center for their invaluable support in pre-processing the neuroimaging dataset.

REFERENCES

[1] S. Agarwal, K. Branson, and S. Belongie, “Higher order learning with graphs,” in Proc. 23rd Int. Conf. Mach. Learn., 2006, pp. 17–24.
[2] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” Rev. Modern Phys., vol. 74, no. 1, 2002, Art. no. 47.
[3] D. Arya, D. K. Gupta, S. Rudinac, and M. Worring, “HyperSAGE: Generalizing inductive representation learning on hypergraphs,” 2020, arXiv: 2010.04558.

[4] D. Arya et al., “Fusing structural and functional MRIs using graph convolutional network for autism classification,” in Proc. 3rd Conf. Med. Imag. Deep Learn., 2020, pp. 44–61.

[5] D. Arya, S. Rudinac, and M. Worring, “HyperLearn: A distributed approach for representation learning in datasets with many modalities,” in Proc. 27th ACM Int. Conf. Multimedia, 2019, pp. 2245–2253.

[6] D. Arya, S. Rudinac, and M. Worring, “Predicting behavioural patterns in.discussion forum posts using deep learning on hypergraphs,” in Proc. Int. Conf. Content-Based Multimedia Indexing, 2019, pp. 1–6.

[7] D. Arya and M. Worring, “Exploiting relational information in social networks using geometric deep learning on hypergraphs,” in Proc. ACM Int. Conf. Multimedia Retrieval, 2018, pp. 117–125.

[8] S. Bai, F. Zhang, and P. H. S. Torr, “Hypergraph convolution and hypergraph attention,” Pattern Recognit., vol. 110, 2021, Art. no. 107637.

[9] A. R. Benson, “Three hypergraph eigenvector centralities,” SIAM J. Math. Data Sci., vol. 1, no. 2, pp. 293–312, 2019.

[10] C. Berge and E. Minieka, Graphs and Hypergraphs. Amsterdam, The Netherlands: North Holland, 1976.

[11] S. Bhagat, G. Cormode, and S. Muthukrishnan, “Node classification in social networks,” in Social Network Data Analytics. Berlin, Germany: Springer, 2011, pp. 155–158.

[12] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: Going beyond Euclidean data,” IEEE Signal Process. Mag., vol. 34, no. 4, pp. 18–42, Jul. 2017.

[13] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” in Proc. Int. Conf. Learn. Representations, 2014, pp. 1–14.

[14] J. Bu et al., “Music recommendation by unified hypergraph: Combining social media information and music content,” in Proc. 18th ACM Int. Conf. Multimedia, 2010, pp. 391–400.

[15] T.-H. H. Chan, A. Louis, Z. G. Tang, and C. Zhang, “Spectral properties of hypergraph Laplacian and approximation algorithms,” J. ACM, vol. 65, no. 3, pp. 1–48, 2018.

[16] I. Chien, C.-Y. Lin, and H.-W. Wang, “Community detection in hypergraphs: Optimal statistical limit and efficient algorithms,” in Proc. Int. Conf. Artif. Intell. Statist., 2018, pp. 871–879.

[17] C. B. Clement, M. Bierbaum, K. P. O’Keeffe, and A. A. Alemi, “On the use of arxiv as a dataset,” 2019, arXiv: 1905.00075.

[18] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, “Principals of graph neural networks,” in Proc. 27th ACM Int. Conf. Multimedia, 2019, pp. 401–414.

[19] J. Bu et al., “Music recommendation by unified hypergraph: Combining social media information and music content,” in Proc. 18th ACM Int. Conf. Multimedia, 2010, pp. 391–400.

[20] T.-H. H. Chan, A. Louis, Z. G. Tang, and C. Zhang, “Spectral properties of hypergraph Laplacian and approximation algorithms,” J. ACM, vol. 65, no. 3, pp. 1–48, 2018.

[21] I. Chien, C.-Y. Lin, and H.-W. Wang, “Community detection in hypergraphs: Optimal statistical limit and efficient algorithms,” in Proc. Int. Conf. Artif. Intell. Statist., 2018, pp. 871–879.

[22] C. B. Clement, M. Bierbaum, K. P. O’Keeffe, and A. A. Alemi, “On the use of arxiv as a dataset,” 2019, arXiv: 1905.00075.

[23] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, “Principals of graph neural networks,” in Proc. 27th ACM Int. Conf. Multimedia, 2019, pp. 401–414.

[24] J. Bu et al., “Music recommendation by unified hypergraph: Combining social media information and music content,” in Proc. 18th ACM Int. Conf. Multimedia, 2010, pp. 391–400.

[25] T.-H. H. Chan, A. Louis, Z. G. Tang, and C. Zhang, “Spectral properties of hypergraph Laplacian and approximation algorithms,” J. ACM, vol. 65, no. 3, pp. 1–48, 2018.

[26] I. Chien, C.-Y. Lin, and H.-W. Wang, “Community detection in hypergraphs: Optimal statistical limit and efficient algorithms,” in Proc. Int. Conf. Artif. Intell. Statist., 2018, pp. 871–879.

[27] C. B. Clement, M. Bierbaum, K. P. O’Keeffe, and A. A. Alemi, “On the use of arxiv as a dataset,” 2019, arXiv: 1905.00075.

[28] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, “Principals of graph neural networks,” in Proc. 27th ACM Int. Conf. Multimedia, 2019, pp. 401–414.
