HEAL: Unlocking the Potential of Learning on Hypergraphs Enriched with Attributes and Layers

Naganand Yadati
National University of Singapore
y.naganand@gmail.com

Tarun Kumar*
Hewlett Packard Enterprise
kumartaruncse@gmail.com

Deepak Maurya
Purdue Unviersity
dmaurya@purdue.edu

Partha Talukdar
Google Research, Bangalore
partha@google.com

Balaraman Ravindran
Robert Bosch Centre for Data Science and AI, IIT Madras
ravi@cse.iitm.ac.in

Abstract

The paper aims to explore the untapped potential of hypergraphs by leveraging attribute-rich and multi-layered structures. The primary objective is to develop an innovative learning framework, Hypergraph Learning Enriched with Attributes and Layers (HEAL), capable of effectively harnessing the complex relationships and information present in such data representations. Hypergraphs offer a more expressive and versatile way to model intricate relationships in real-world systems, accommodating entities with multiple interactions and diverse attributes. However, existing learning methods often overlook these unique features, especially cross-layer interactions, hindering their full potential. The motivation behind this research is to bridge this gap by creating HEAL, a novel learning approach that capitalises on attribute-rich and multi-layered hypergraphs to achieve superior performance across various applications. HEAL adopts a feature smoothing strategy to propagate attributes over the hypergraph structure, enabling the decoupling of feature propagation and transformation steps. This innovative methodology allows HEAL to capture the intricacies of multi-layer interactions while efficiently handling attribute-rich data. Moreover, the paper presents a detailed analysis of HEAL's design and performance, showcasing its effectiveness in handling complex real-world datasets. The implications of HEAL are far-reaching and promising. By unlocking the potential of learning on hypergraphs enriched with attributes and layers, our work opens up new possibilities in various domains. This research contributes to the advancement of graph-based learning methods, paving the way for more sophisticated and efficient approaches in real-world applications.

1 Introduction

Network representation learning (NRL) is an active area of research focused on learning vertex representations in a low-dimensional space while preserving the input network properties. NRL techniques have been developed for a wide range of network types, including signed graphs, multi-layer graphs, and multi-networks. Although effective, existing NRL techniques on these structures are limited to handling pairwise relationships. Real-world datasets, such as movie databases, academic data, social networks, and product review data, present a higher level of complexity, where vertex relationships extend beyond pairwise associations and can be more appropriately modelled using hypergraphs. For instance, hypergraphs allow us to capture all movies involving a cast member in movie databases, e.g., actor, director, or all documents co-authored by an author in academic datasets.

*work done while at IIT Madras.

Yadati et al., HEAL: Unlocking the Potential of Learning on Hypergraphs Enriched with Attributes and Layers. *Proceedings of the Second Learning on Graphs Conference (LoG 2023)*, PMLR 231, Virtual Event, November 27–30, 2023.

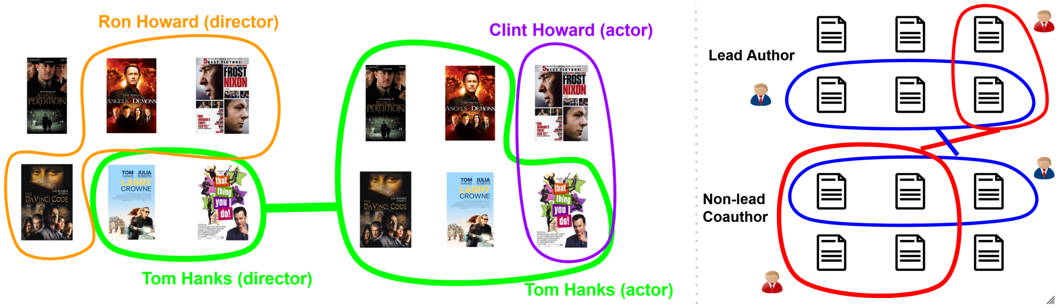


Figure 1: (Best seen in colour) Movie (left) and academic multi-layer hypergraphs (right). Within the movie database, the vertices represent movies, with 6 movies showcased in this specific example. Hyperedges connect movies sharing a common cast member (e.g., actor). Layers represent different cast members such as directors and actors. Cross-layer relationships represent actor-cum-director relationships (e.g., Tom Hanks). Attributes represent movie content such as plot key words and movie title. In the academic image, also containing six vertices representing documents, hyperedges represent authors. Cross-layer relationships represent the same author appearing as a lead author in certain documents and as a non-lead co-author in others. Attributes are obtained through the contents such as title and abstract. Please see Section 1 for details.

Furthermore, real-world data exhibits multi-layered [2] relationships. Previous studies have explored higher-order and multi-layer relationships independently. In contrast, our paper proposes a novel joint modelling of these intricate relationships (higher-order and multi-layer) in real-world data using a multi-layer hypergraph. The incorporation of joint modelling not only extends the applicability of existing structures but also reveals new types of relationships, such as cross inter-layer connections, in real-world data. These relationships mirror factual occurrences, for instance, the same cast member appearing as an actor in one set of movies and as a director in a potentially different set of movies in movie databases. In academic datasets, the same author may undertake lead authorship in certain documents and non-lead co-authorship in others. Figure 1 illustrates these examples.

While existing NRL techniques can leverage higher-order and multi-layer relationships, their adaptation to accommodate these novel relationship types in multi-layer hypergraphs remains unclear. Our contributions aim to bridge these gaps and can be summarised as follows:

1. We formally define the novel problem of learning representations on an attributed multi-layer hypergraph, which offers a generic and powerful representation for real-world networks. In particular, it generalises attributed graph, hypergraph, and multi-layer network representation learning problems. **The first technical innovation** is that our work introduces a novel way to represent real world networks, capturing attribute-rich information as well as higher-order, multi-layer, and, most significantly, cross-layer interactions. This representation enables more informative representations for vertices and hyperedges.

2. We propose and develop HEAL (Hypergraph Learning Enriched with Attributes and Layers), employing an advanced attribute smoothing strategy to effectively propagate attributes across the multi-layer hypergraph structure. **The second technical innovation** is the integration of cross [3] higher-order inter-layer connections in the proposed smoothing strategy.

3. The experimental findings provide strong evidence of HEAL's effectiveness in successfully addressing vertex classification and hyperedge prediction tasks. In our rigorous evaluation, we benchmarked HEAL against several representative models from four distinct domains: graph, hypergraph, multi-layer graph, and heterogeneous graph models.

[2] When we use the term 'layer,' we are referring to the layers within the dataset context, rather than the conventional hidden layers in a neural network.

[3] In multi-layer networks, a cross connection connects vertex i of layer l_1 to any vertex j in layer l_2. Here, i and j need not be the same.

2 Related Work

In this section, we set the context for our research and conduct a review of the relevant literature to frame our study. We categorise the relevant literature into distinct groups, recognising that the categories may intersect, to offer readers a systematic review.

Network Representation Learning (NRL) is a field of machine learning that aims to learn low-dimensional vector representations of network vertices that capture their structural and semantic features. Some of the most popular NRL methods include DeepWalk [1], Node2Vec [2], LINE [3], and SDNE [4]. NRL techniques have been applied to a wide range of information-rich structures, including but not limited to signed networks [5], directed networks [6, 7], attributed networks [8], multi-networks [9], and dynamic networks [10]. Some of the most recent areas of focus in NRL include quantum embeddings [11], scalability for large dynamic networks [12], and out-of-distribution generalisation in directed networks [13].

Graph Neural Networks (GNNs) belong to a class of deep neural networks specialised in handling irregular graph data. Popular GNNs include GCN [14], GAT [15], GraphSAGE [16], and GIN [17]. Books [18–21] and detailed surveys [22–24] provide a comprehensive account of the significant advancements in the areas of GNNs and NRL. *Decoupled Graph Neural Networks* have proven to be effective models in Graph Machine Learning tasks, where the feature propagation mechanism and feature transformation mechanism are separated [25–32]. The compelling evidence from decoupled GNNs and simpler models, such as SGC on homophilic data [33], along with its heterophilic variant [34], and spectral variant [35], highlights the significance of feature smoothing as a crucial mechanism. For these reasons, *we focus on devising a feature-smoothing-based approach in our work*. Advanced coupled deep learning-based methods are left for future investigation.

Embedding Multiplex Networks. Traditionally, multiplexity in networks has been employed to capture diverse aspects of social relationships among users [36], communication frequencies in distinct brain regions [37], and various modes of transportation among cities [38]. In the realm of multiplex network embedding, researchers have taken inspiration from traditional NRL techniques to represent the intricate relationships between vertices across different layers [39, 40]. Several recent studies emphasise the use of semi-supervised learning (and self-supervised learning) techniques to extract vertex embeddings without external supervision, maximising mutual information between vertex embeddings and graph summaries [41–43]. Several deep learning approaches have been introduced, employing/extending GCNs for vertex representations in multiplex networks [44–46]. *In contrast to prior research on multiplex networks*, we introduce a novel feature-smoothing-based methodology for learning on multi-layer hypergraphs.

Learning on Hypergraphs. The weighted clique expansion, first introduced in a seminal work [47], has emerged as a widely studied technique for approximating hypergraphs as graphs [48, 49]. Novel developments in the area of non-uniform hypergraphs have emerged, leading to more effective exploitation of the hypergraph structure through non-linear models [50–54]. The essence of HGNN [55] lies in its transformation of the hypergraph into a weighted clique and its utilisation of message passing strategies akin to GCNs. By using a non-linear Laplacian, HyperGCN [56] establishes connections only between the most discrepant vertices and extends the connections to mediator vertices in each hyperedge. In recent times, akin to the trend observed in GNNs, attention models have gained popularity within the hypergraph learning domain [57–63]. G-MPNN [64] is a generalised message passing neural network for multi-relational hypergraphs which works best for vertices appearing in a fixed order in each hyperedge. The prevailing trends in this line of research comprise set-based methods [65], equivariant models [66, 67], and energy functionals [68]. Very recently, there has been growing attention towards the topic of topological deep learning [69, 70], focusing on topological domains that encompass hypergraphs, as well structures like simplicial complexes and cell complexes.

Increasing the complexity of a hypergraph to a multi-layer network can be achieved by introducing multiple layers and inter-layer edges, where vertices in these edges correspond to the same real-world entity. Surprisingly, *none of the existing studies in both the hypergraph and multi-layer network literature consider the presence of inter-layer variable vertex hyperedges*, e.g., actor-cum-director relationships in movie databases and lead-coauthor relationships in academic data as shown in Figure 1. This paper addresses this critical limitation in prior research and sheds light on the significance of such inter-layer hyperedges in real-world relational data in the context of representation learning.

Table 1: Example methods on networks. Please see Section 2 for details. L-wise: Layer-wise.

Recent methods	Input data	Multi-layer	Hypergraph	Cross relations	L-wise attributes
GCN [14], SGC [33], GAT [15]	Undirected graph	×	×	×	×
HGNN [55], HyperGCN [56], AllSet [65], UniGNN [60], ED-HNN [66]	hypergraph	×	✓	×	×
SSDCM [43], HDMI [42] DMGI [41]	Multi-layer graph	✓	×	×	×
HEAL (this paper)	Multi-layer hypergraph	✓	✓	✓	✓

Deep Learning on Heterogeneous Graphs. Due to the prevalence of multi-typed real-world objects and interactions, recent investigations, summarised in surveys [71–73], have examined the extension of deep learning methods on graphs to cater to the needs of heterogeneous graphs. The core concept behind these methods is to utilise vertex types, edge types, and meta-path semantics for effective projection and aggregation through attention mechanisms [74–78]. Another closely related research area focuses on multi-relational graphs, such as knowledge graphs, with a primary emphasis on effectively managing numerous relation types [79–83].

While heterogeneous graphs provide a feasible approach to model all the real-world datasets examined in this paper, we present an alternative approach of modeling the datasets as multi-layer hypergraphs. *This approach renders some of the novel ideas explored in the paper much more natural and evident*, because of grouping relationships of hyperedges and cross-layer variable vertex interactions.

3 Technical Challenges and Novel Problem Definition

The challenges involved in the work are as follows:

1. Topology Definition: Defining the topological structure for cross higher-order inter-layer connections can be complex, particularly when the network has diverse relationships and attribute-rich data. Ensuring the connections are meaningful and relevant to the problem at hand is crucial.
2. Data Complexity: Representing real-world networks with attribute-rich information, higher-order relationships, and multi-layer structures can lead to increased data complexity. Handling and processing such data efficiently require innovative techniques.
3. Efficient Integration of Cross-Layer Interactions: Capturing cross-layer interactions in multi-layer hypergraphs presents challenges in defining appropriate connections between different layers and handling attributes of variable dimensions across layers. Integrating cross higher-order inter-layer connections requires efficient algorithms.

Within this section, we introduce various network types relevant to our study and tackle challenge 1 by formulating a novel problem of learning representations on a multi-layer hypergraph framework. Challenges 2 and 3 are then addressed in the subsequent section.

Definition 1 *(**Directed hypergraph** [84]). An ordered pair $H = (V, E)$, where V is a vertex set, and $E = \Big\{(t_1, h_1), \cdots, (t_m, h_m)\Big\} \in 2^V \times 2^V$ is a set of directed hyperedges. Each $e \in E$ is of form (t, h), $t \subseteq V$ is a tail and $h \subseteq V$ is a head hyperedge with $t \neq \emptyset$ and $h \neq \emptyset$.*

Definition 2 *(**Multi-layer graph** [85]). A triple (V, E, L) where V is a set of n vertices, L is a set of layers, and $E = \Big\{(u, w, l, \ell) : u, w \in V, \text{ and } l, \ell \in L\Big\} \subseteq V \times V \times L \times L$ is a set of multi-layer edges.*

Here, (u, w, l, ℓ) encodes an edge from a vertex u of layer l to a vertex v of layer ℓ. The multi-layer graph is undirected whenever $(u, w, l, \ell) \in E \implies (w, u, , \ell, l) \in E$ for each $(u, w, l, \ell) \in E$; otherwise it is directed.

Definition 3 *(**Attributed graph**). A triple (V, E, X) where V is a set of n vertices, E is a set of edges, and the set $X = \Big\{x_v \in \mathbb{R}^d : \ v \in V\Big\}$ is a set of d-dimensional attributes for the vertices.*

Each $v \in V$ is associated with a d-dimensional vector $x_v \in \mathbb{R}^d$. X is represented by an $n \times d$ matrix. We now define the network type that we study in this paper.

Definition 4 ***Attributed Multi-layer hypergraph****.*
A quadruple $\mathcal{H} = (V, E, X, L)$ where V is a set of n vertices, L is a set of layers,

$$E = \{(t, h, l, \ell) : \; t, h \subseteq V, \text{ and } l, \ell \in L\} \subseteq 2^V \times 2^V \times L \times L \tag{1}$$

is set of multi-layer directed hyperedges. $X = \left\{X^{(l)} = \{x_v^{(l)} \in \mathbb{R}^{d_l} : \; v \in V\}\right\}_{l \in L}$ is a set of layer-specific attributes for the vertices $v \in V$. We can also write X as a set of $|L|$ matrices, $X = \{X^{(1)}, \cdots, X^{(|L|)}\}$, $X^{(l)} \in \mathbb{R}^{n \times d_l}$ for $l = 1, \cdots, |L|$.

For a movie database with actor-cum-director relationships, we assume $(t, h, l, \ell) \in E \implies (h, t, \ell, l) \in E$ for all $e \in E$.. A similar assumption is made for lead-cum-non-lead authors in academic datasets. The assumptions made are analogous to those in undirected graphs, where edges are bidirectional and symmetric in nature. However, directed hypergraphs provide a unique approach to indicate the presence of pairwise relationships between hyperedges.

By adopting directed hyperedges, we can effectively encode asymmetrical relationships present in cross-layer connections, enhancing the flexibility of our modelling approach. For example, directions could indicate the existence of an asymmetrical relationship from actors to directors or lead authors to co-authors. We leave exploration of asymmetrical relationships for future investigation.

We note that $\mathcal{H}$ is quite generic as it generalises

- Attributed directed hypergraph if $|L|$ =1.
- Attributed directed multi-layer graph if $|t| = |h| = 1$, $l = \ell$ for $(t, h, l, \ell) \in E$
- Attributed directed graph if $|t| = |h| = 1$ for $(t, h, l, \ell) \in E$ and L is singleton.

Problem 1 ***Attributed Multi-layer hypergraph Representation Learning****.*
Given $\mathcal{H} = (V, E, X, L)$, the problem is to produce a unified low-dimensional space embedding of each vertex $v \in V$ on every layer $l \in L$. The goal is to find a function $f_\Theta : V \rightarrow \mathbb{R}^r$ with parameters Θ to learn meaningful vertex representation vectors where $r \ll n$.

4 HEAL: Hypergraph Learning Enriched with Attributes and Layers

In this section, we present HEAL to compute vertex representations within multi-layer hypergraphs.

4.1 Intuition

Within our multi-layer hypergraph structure, proposed in Definition 4, three fundamental building blocks come into play: multiple layers, hyperedges, and inter-layer connections that interlink pairs of hyperedges from (possibly) different layers. The essence of our approach hinges on the intricate propagation of features across hyperedges and inter-layer connections, culminating in the fusion of representations from multiple layers into a singe vertex representation.

Our innovative approach inherently incorporates three distinct modules, designed to handle the three building blocks. A key insight shaping our approach is that a directed hypergraph can be conceptualised as a fusion of two structures: a hypergraph representing vertex relationships superimposed with a secondary graph structure of pairwise relationships between hyperedges.

4.2 Motivation

A logical design strategy involves picking appropriate methodologies for the hypergraph structure, like utilising a neural network on hypergraphs, and for the secondary graph structure concerning hyperedges, like utilising a graph neural network. Recent studies have highlighted that the effectiveness of neural networks on most real-world relational structures can be attributed to a decoupled process of feature smoothing [25–32].

One significant benefit of employing feature smoothing compared to intricate coupled neural networks lies in its adaptability to the specific vertex under consideration. For instance, vertices possessing weaker neighborhood structures can gain advantages from an extended range of smoothing. The subsequent subsections delve into the mechanics of propagating features throughout the attributed multi-layer hypergraph structure. A model diagram is visually presented in Figure 3, a vertex's perspective in Figure 4, and an algorithm is presented in Algorithm 1.

4.3 Handling Inter-layer Relationships

An inter-layer relationship takes the form of a quadruple denoted as (t, h, l, ℓ). To facilitate the feature propagation process, we initiate the process by assigning one-hot attributes to all vertex sets t and h, to encode the respective layer details l and ℓ. We call the resulting attribute matrix $X_e \in \mathbb{R}^{m \times |L|}$ where $m = |E|$. Assuming that the adjacency with self-loops connecting the hyperedges is $A \in \mathbb{R}^{m \times m}$, we use a Graph Feature Propagator (GFP) on the adjacecny A and X_e, i.e., $\tilde{X}_e = \text{GFP}(A, X_e)$.

4.4 Handling the Hypergraph Structures

A hypergraph can be fully encoded using a rectangular binary matrix of incidence, indicating the presence or absence of vertices in each hyperedge. We define the incidence matrices that encode the hypergraph structures for all layers as $I^{(l)}$, where l ranges from 1 to $|L|$. We propose a novel hypergraph feature propagator, HFP, to propagate attributes over the hypergraph structures. The updated vertex representations can be summarised as $\tilde{X}_v^{(l)} = \text{HFP}\Big(I^{(l)}, X_v^{(l)}, \tilde{X}_e\Big)$.

Details of HFP. In the present landscape of hypergraph methodologies, a prominent trend involves tightly coupled approaches on the hypergraph star expansion [58, 60, 63, 65, 66, 68]. Motivated by this, we propose to propagate vertex and hyperedge features over the hypergraph star expansion.

Our initial step involves bringing $X_v^{(l)}$ and $\tilde{X}_e$ into a shared vector space via zero padding. To be precise, considering the total dimensionality as $d_l + |L|$, we append zero vectors of size $|L|$ to row vectors in $X_v^{(l)}$ and append row vectors in $\tilde{X}_e$ to zero vectors of size d_l. The matrices obtained are denoted as $\bar{X}_v^{(l)}$ and $\bar{X}_e^{(l)}$ correspondingly, emphasising that the introduction of d_l zeroes has transformed the initially layer-independent X_e into a layer-dependent matrix.

Following this, the subsequent phase of HFP encompasses an iterative process of smoothing vertices and hyperedges in a mutual manner. To provide an instance of a smoothing method, take the case of degree-based averaging, depicted as $X_v = D_v^{-1} I \bar{X}_e$ and $X_e = D_e^{-1} I^T \bar{X}_v$.

Advantages. A distinctive benefit of feature smoothing, as opposed to tightly coupled neural networks, lies in its ability to accommodate vastly distinct iterations of smoothing for vertices and hyperedges. This adaptability is governed by a dataset-specific smoothing parameter denoted as τ. The number of HFP iterations is chosen to ensure that the smoothed features $\tilde{\mathbf{x}}$ closely approximate the original features $\bar{\mathbf{x}}$, so that $\min\{i_v : ||\tilde{\mathbf{x}} - \bar{\mathbf{x}}||_2 \leq \tau\}$ for all vertices. Here i_v is the number of smoothing iterations for vertex v. Please see Algorithm 1 for details.

The issue of oversmoothing is prevalent in tightly coupled neural networks on hypergraphs [60, 66], as these networks, limited by a fixed number of hidden layers , are constrained to gather information from an equal number of hops around each vertex. In order to get around concerns of oversmoothing, we conduct comparisons with the top-performing baseline models. The examination of oversmoothing under different iterations of smoothing is an interesting topic to explore in future.

4.5 Fusing Multiple Layers

After acquiring smoothed features $\tilde{X}_v^{(l)}$ through HFP, where l ranges from 1 to $|L|$, our aim in this subsection is to merge them into a unified representation for subsequent tasks. Since the dimension of layer l is $d_l + |L|$, our proposal presents $|L|$ feed-forward neural networks (multi-layer perceptrons) with a shared hidden representation space. In other words, if $H_v^{(l)}$ is the hidden representation of vertex v, we obtain $H_v^{(l)} = MLP_l(\tilde{X}_v^{(l)})$ for $l = 1, \cdots, |L|$ and all $v \in V$ and then use an attention mechanism to fuse $H_v^{(l)}, l = 1, \cdots |L|$ to get the hidden representation for a vertex.

4.6 Training

To perform multi-class vertex classification involving c classes, we employ a linear layer to map H_v onto a softmax output layer, and the model is trained using cross-entropy loss. To conduct multi-label vertex classification encompassing c labels, we utilize sigmoid activation on each dimension of the linear layer's output and the training process involves the use of binary cross-entropy loss. For hyperedge prediction (more details in the appendix), we use a ranking objective.

Table 2: Vertex classification on movie datasets. Please see Section 5.1 for details.

Model Type	Method	IMDB-MC		IMDB-ML	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1
Graph Agnostic	MLP	59.97±0.72	60.47±0.42	46.93±1.02	50.03±0.87
Graph	GCN [14]	64.84±0.82	64.21±0.47	47.93±1.33	50.82±0.64
	SGC [33]	64.74±0.54	64.98±0.27	45.83±0.24	50.74±0.05
	GAT [15]	65.94±0.87	65.79±0.58	47.32±1.26	52.86±0.63
Hypergraph	HGNN [55]	64.84±0.88	65.11±0.49	46.02±1.34	50.13±0.93
	HyperGCN [56]	64.98±1.62	64.21±0.97	45.97±2.65	50.22±1.15
	UniGNN [60]	64.94±0.58	65.00±0.39	50.53±1.35	54.22±1.03
	AllSet [65]	65.13±0.73	64.93±0.41	48.91±0.96	53.16±0.65
	ED-HNN [66]	64.99±0.81	65.76±0.63	47.98±1.42	52.49±0.93
Multi-layer Graph	DMGI [41]	65.76±0.92	65.56±0.43	49.21±1.29	52.76±0.49
	HDMI [42]	65.39±0.86	64.96±0.50	49.47±1.21	53.68±0.44
	SSDCM [43]	67.79±0.73	67.83±0.45	52.04±1.19	55.21±0.79
Heterogeneous Graph	RGCN [83]	66.02±0.59	66.17±0.41	49.98±1.26	53.75±0.73
	MAGNN [76]	66.67±0.99	66.45±0.62	50.22±1.35	53.98±0.58
	SHGN [74]	67.54±0.82	67.39±0.58	51.76±1.12	55.62±0.68
Proposed Model	**HEAL**	**69.14±0.87**	**69.46±0.54**	**53.59±1.18**	**57.88±0.62**

4.7 Computational Complexity Analysis

Let R be the maximum number of smoothing iteration of HFP of all the vertices $v \in V$. Let M be the maximum number of vertex-hyperedge pairs in hypergraphs across all layers and $m = |E|$. The time complexity of smoothing is $\mathcal{O}(R(M + m)d)$ where d is the maximum number of input features across all layers. The time complexity of training and inference is $\mathcal{O}(nd^2)$ time where $n = |V|$ is the number of vertices in G, assuming the number of hyperedges is linear in the number of vertices.

5 Experiments

In this section, we provide a thorough analysis of experiments that confirm the efficacy of HEAL.

Overview. We present a baseline comparison for node classification in both Table 2 and Table 4, along with an analysis of hyperedge prediction performance outlined in Table 5. We systematically analyse the components of HEAL through an ablation study in Table 3. We visualise embeddings of HEAL and competitive baselines in Figure 2. To further showcase the effectiveness of HEAL, we conduct supplementary experiments, encompassing variations in train split proportions as depicted in Table 11, random splits as detailed in Table 6, hyperparameter analyses in Section E, running time tradeoffs in Figure 9, and exploration of potential FP candidates as elaborated in Table 12.

Datasets. Two movie database datasets and two academic datasets are employed in our vertex classification tasks. Additionally, in the appendix specifically in Table 5, we evaluate hyperedge prediction performance on three more datasets. Please see Section F for details.

Training and Inference Details. For vertex classification, we utilize both Micro and Macro averaged F1 scores as our evaluation metrics. For hyperedge prediction, we report the ROC-AUC, Average Precision, and Precision@K. For HEAL's configuration, please refer to Section E.

5.1 Baseline Comparison

To ensure a comprehensive evaluation, we consider baselines from four separate research directions, namely graph, hypergraph, multi-layer graph, and heterogeneous graph. We then select representative methods from each category as baselines for comparison. Please see Section G for a detailed description of baselines.

Table 3: Ablation study on movie datasets. Please see Section 5.2 for details.

Method	IMDB-MC		IMDB-ML	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
MLP on Vertex Features	59.97±0.72	60.47±0.42	46.93±1.02	50.03±0.87
Without GFP and HFP	61.12±0.64	61.98±0.40	48.04±1.08	51.11±0.84
Without HFP	61.60±0.69	62.45±0.49	48.23±0.97	51.47±0.74
Without GFP	68.06±1.01	68.20±0.73	51.87±1.11	55.37±0.82
HEAL with All Layers Unified	67.95±0.81	67.98±0.41	52.04±1.23	56.36±0.59
HEAL	**69.14±0.87**	**69.46±0.54**	**53.59±1.18**	**57.88±0.62**

Observations. Table 2 and Table 4 present vertex classification results on the movie datasets and authorship datasets respectively. In a general sense, our suggested HEAL consistently outperforms all baseline methods across commonly used metrics. The superior performance of HEAL can be ascribed to its proficiency in capturing advanced higher-order interactions and inter-layer connections.

5.2 Ablation Study

Within this section, we provide a breakdown of the ablation analysis performed on HEAL in order to assess the effects of its individual components. HEAL comprises three primary phases: firstly, handling inter-layer connections via the graph feature propagator (GFP); secondly, handling the hypergraph structures using the hypergraph feature propagator (HFP); and finally, fusing the layers.

In this context, we have identified the following four ablated baselines for HEAL:

- Without GFP and HFP: This is a graph-agnostic approach that preserves the component assigning an MLP to each layer, utilising a common representation for every vertex.
- Without HFP: This baseline exclusively relies on inter-layer pairwise connections, disregarding the hypergraph structures present within each layer.
- Without GFP: In this baseline, the inter-layer connections are disregarded, and the reliance is on harnessing the hypergraph structures across all layers.
- HEAL with All Layers Unified: In this baseline, the element that assigns an MLP to each hypergraph layer is excluded. Additionally, the baseline takes the unioin of all hyperedges from every layer. This is essentially applying HEAL to a single-layer hypergraph.

Observations. Table 3 presents the findings of the ablation study performed on movie datasets. From the table, it is evident that the hypergraph feature propagator (HFP) component stands out as the most crucial aspect of HEAL. Furthermore, the table illustrates that the performance of HEAL relies on the collective contribution of all its components.

5.3 Embedding Visualisation

Figure 2 shows the t-SNE visualisations of HEAL (leftmost) and five competitive baselines. The color code indicates the vertex classes in the CoraLCA dataset. We selected the vertex embeddings that yielded the highest node classification scores across all competing methods. It is evident that all methods produce interpretable visualisations, demonstrating distinct inter-class separation. HEAL achieves compact, well-separated clusters with the same class labels, exhibiting improved separation.

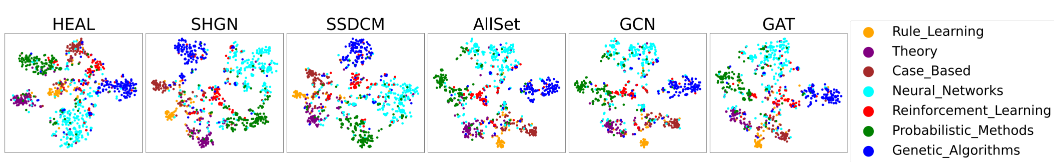


Figure 2: t-SNE visualisation on CoraLCA. Please see Section 5.3 for details

Table 4: Node classification on author datasets. Please see Section 5.1 for details.

Model Type	Method	CoraLCA		DBLP-LCA	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1
Graph Agnostic	MLP	58.14±0.81	61.36±0.78	63.48±0.74	67.28±0.76
Graph	GCN [14]	67.98±0.57	72.17±0.64	72.03±0.72	76.14±0.69
	SGC [33]	66.53±0.25	70.96±0.21	70.63±0.19	74.03±0.35
	GAT [15]	67.47±0.65	72.04±0.68	72.09±0.76	75.83±0.68
Hypergraph	HGNN [55]	66.93±0.65	70.85±0.63	70.61±0.77	75.04±0.88
	HyperGCN [56]	66.79±1.26	70.91±1.46	70.65±1.03	74.27±0.83
	UniGNN [60]	67.03±0.65	71.03±0.49	70.87±0.65	74.70±0.72
	AllSet [65]	66.76±0.61	71.01±0.56	70.49±0.82	74.86±0.63
	ED-HNN [66]	66.60±0.86	70.79±0.60	70.62±0.66	74.72±0.86
Multi-layer Graph	DMGI [41]	67.44±0.61	70.97±0.65	70.57±0.64	74.19±0.74
	HDMI [42]	67.28±0.73	71.13±0.63	70.40±0.67	74.62±0.60
	SSDCM [43]	68.62±0.67	72.04±0.58	72.16±0.60	76.55±0.71
Heterogeneous Graph	RGCN [83]	66.59±0.67	70.83±0.54	70.71±0.89	74.92±0.77
	MAGNN [76]	66.11±1.03	70.24±0.82	69.39±0.75	72.80±0.72
	SHGN [74]	68.73±0.64	72.58±0.62	72.59±0.69	76.98±0.65
Proposed Model	**HEAL**	**69.81±0.62**	**73.34±0.59**	**73.79±0.71**	**78.47±0.68**

6 Future Work and Concluding Remarks

There are numerous possibilities for extending and enhancing our work including:

Theoretical Exploration of Propagation Mechanisms. Future research can delve into utilizing tools from Markov Processes and Spectral Graph Theory to set an upper bound on the number of propagation iterations in HFP. This approach, informed by factors such as node degree and the second eigenvalue of the normalized adjacency matrix, offers promising directions for theoretical analysis. Our paper, while empirically focused on multi-layer hypergraph layers, suggests the need for theoretical investigations into feature propagation mechanisms like HFP and GFP.

Overlapping Hyperedges. Exploring the connection of overlapping hyperedges, especially when sharing a single vertex or multiple vertices, offers intriguing research potential. While not addressed in this study, this area could enhance our understanding of complex network structures.

Neural Network Architectures. Our method's core feature smoothing strategy can be integrated into GCN-style architectures, enhancing neural network models. Incorporating text encoding alongside relational data modeling, especially in multi-layer hypergraphs, is a promising avenue, exemplified by recent work [86] and a promising future research direction.

Networks with Sensitive or No Attributes. Exploring scenarios with missing or sensitive attributes, like users hiding certain information in social networks, remains underexplored beyond traditional graph data. Addressing these complexities not only benefits multi-layer hypergraphs but also extends to other domains.

Scalability. Adapting our approach to large datasets presents challenges due to memory limitations, especially when identifying layered hyperedges and cross-layer connections. Identifying datasets with diverse entity interactions and group relationships can assist, although scaling neural networks on multi-layer hypergraphs might face significant memory constraints.

Conclusion

We present an innovative approach to model real-world networks, encompassing not only attribute-rich data but also higher-order structures, multi-layer relationships, and notably, cross-layer interactions. A distinctive contribution of our work lies in the incorporation of cross higher-order inter-layer connections within the proposed smoothing strategy. Through this novel framework, we aim to advance the field of network analysis and foster new avenues of research into the intricate interplay of attributes and interactions within diverse systems.

Acknowledgements

This work was partially supported by the Ministry of Education, Government of India. This work was also partially supported by Intel research grant RB/18-19/CSE/002/INTI/BRAV to BR.

References

[1] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 701–710, 2014. 3

[2] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 855–864, 2016. 3

[3] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large info-network embedding. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 1067–1077, 2015. 3

[4] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1225–1234, 2016. 3

[5] Suhang Wang, Charu Aggarwal, Jiliang Tang, and Huan Liu. Attributed signed network embedding. In *Proceedings of the ACM on Conference on Information and Knowledge Management (CIKM)*, pages 137–146, 2017. 3

[6] Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang. Side: Representation learning in signed directed networks. In *Proceedings of the World Wide Web Conference (WWW)*, pages 509–518, 2018. 3

[7] Yiqi Chen, Tieyun Qian, Huan Liu, and Ke Sun. "bridge": Enhanced signed directed network embedding. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 773–782, 2018. 3

[8] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. Network representation learning with rich text information. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*, pages 2111–2117, 2015. 3

[9] Jingchao Ni, Shiyu Chang, Xiao Liu, Wei Cheng, Haifeng Chen, Dongkuan Xu, and Xiang Zhang. Co-regularized deep multi-network embedding. In *Proceedings of the World Wide Web Conference (WWW)*, pages 469–478, 2018. 3

[10] Lun Du, Yun Wang, Guojie Song, Zhicong Lu, and Junshan Wang. Dynamic network embedding : An extended approach for skip-gram based network embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2086–2092, 2018. 3

[11] Xinyu Ye, Ge Yan, and Junchi Yan. Vqne: Variational quantum network embedding with application to network alignment. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, page 3105–3115, 2023. 3

[12] Haoran Deng, Yang Yang, Jiahe Li, Haoyang Cai, Shiliang Pu, and Weihao Jiang. Accelerating dynamic network embedding with billions of parameter updates to milliseconds. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, page 414–425, 2023. 3

[13] Hyunsik Yoo, Yeon-Chang Lee, Kijung Shin, and Sang-Wook Kim. Disentangling degree-related biases and interest for out-of-distribution generalized directed network embedding. In *Proceedings of the ACM Web Conference (TheWebConf)*, page 231–239, 2023. 3

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 3, 4, 7, 9, 18, 24

[15] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. 3, 4, 7, 9, 18, 24

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS) 30*, pages 1024–1034. Curran Associates, Inc., 2017. 3

[17] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019. 3

[18] Lingfei Wu, Peng Cui, Jian Pei, and Liang Zhao. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer Singapore, 2022. 3

[19] Yao Ma and Jiliang Tang. *Deep Learning on Graphs*. Cambridge University Press, 2020.

[20] William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

[21] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *Computing Research Repository (CoRR)*, abs/2104.13478, 2021. 3

[22] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 4–24, 2021. 3

[23] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74, 2017.

[24] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process.*, 34(4):18–42, 2017. 3

[25] Keke Huang, Jing Tang, Juncheng Liu, Renchi Yang, and Xiaokui Xiao. Node-wise diffusion for scalable graph learning. In *Proceedings of the ACM Web Conference (TheWebConf)*, pages 1723–1733, 2023. 3, 5

[26] Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, and Peng Cui. On the equivalence of decoupled graph convolution network and label propagation. In *Proceedings of The Web Conference (TheWebConf)*, pages 3651–3662, 2021.

[27] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin Benson. Combining label propagation and simple models out-performs graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2021.

[28] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations (ICLR)*, 2021.

[29] Wentao Zhang, Mingyu Yang, Zeang Sheng, Yang Li, Wen Ouyang, Yangyu Tao, Zhi Yang, and Bin CUI. Node dependent local smoothing for scalable graph learning. In *Advances in Neural Information Processing Systems (NeurIPS) 34*, pages 20321–20332. Curran Associates, Inc., 2021.

[30] Ming Chen, Zhewei Wei, Bolin Ding, Yaliang Li, Ye Yuan, Xiaoyong Du, and Ji-Rong Wen. Scalable graph neural networks via bidirectional propagation. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, pages 14556–14566. Curran Associates, Inc., 2020.

[31] Aleksandar Bojchevski, Johannes Klicpera, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemberczki, Michal Lukasik, and Stephan Günnemann. Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2464–2473, 2020.

[32] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations (ICLR)*, 2019. 3, 5

[33] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 6861–6871, 2019. 3, 4, 7, 9, 24

[34] Sudhanshu Chanpuriya and Cameron Musco. Simplified graph convolution with heterophily. In *Advances in Neural Information Processing Systems (NeurIPS) 35*, pages 27184–27197. Curran Associates, Inc., 2022. 3

[35] Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[36] Lois Verbrugge. Multiplexity in adult friendships. *Social Forces*, pages 1286–1309, 1979. 3

[37] Manlio De Domenico. Multilayer modeling and analysis of human brain networks. *Giga Science*, 6(5):gix004, 2017. 3

[38] Riccardo Gallotti and Marc Barthelemy. The multilayer temporal network of public transport in britain. *Scientific data*, 2015. 3

[39] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. Scalable multiplex network embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3082–3088, 2018. 3

[40] Jundong Li, Chen Chen, Hanghang Tong, and Huan Liu. Multi-layered network embedding. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 684–692, 2018. 3

[41] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. Unsupervised attributed multiplex network embedding. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 5371–5378, 2020. 3, 4, 7, 9, 18, 24

[42] Baoyu Jing, Chanyoung Park, and Hanghang Tong. HDMI: High-order deep multiplex infomax. In *Proceedings of The Web Conference (TheWebConf)*, pages 2414–2424, 2021. 4, 7, 9, 18, 19, 24

[43] Anasua Mitra, Priyesh Vijayan, Ranbir Sanasam, Diganta Goswami, Srinivasan Parthasarathy, and Balaraman Ravindran. Semi-supervised deep learning for multiplex networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1234–1244, 2021. 3, 4, 7, 9, 24

[44] Yao Ma, Suhang Wang, Charu Aggarwal, Dawei Yin, and Jiliang Tang. Multi-dimensional graph convolution networks. In *SIAM International Conference on Data Mining (SDM)*, pages 657–665, 2019. 3

[45] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1358–1368, 2019.

[46] Hansheng Xue, Luwei Yang, Vaibhav Rajan, Wen Jiang, Yi Wei, and Yu Lin. Multiplex bipartite network embedding using dual hypergraph convolutional networks. In *Proceedings of The Web Conference (TheWebConf)*, pages 1649–1660, 2021. 3

[47] Denny Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Proceedings of the 19th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 1601–1608, 2007. 3

[48] Sameer Agarwal, Kristin Branson, and Serge Belongie. Higher order learning with graphs. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 17–24, 2006. 3

[49] Sai Nageswar Satchidanand, Harini Ananthapadmanaban, and Balaraman Ravindran. Extended discriminative random walk: A hypergraph approach to multi-view multi-relational transductive learning. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*, pages 3791–3797, 2015. 3

[50] Matthias Hein, Simon Setzer, Leonardo Jost, and Syama Sundar Rangapuram. The total variation on hypergraphs - learning on hypergraphs revisited. In *Advances in Neural Information Processing Systems (NeurIPS) 25*, page 2427–2435. Curran Associates, Inc., 2013. 3

[51] Chenzi Zhang, Shuguang Hu, Zhihao Gavin Tang, and T-H. Hubert Chan. Re-revisiting learning on hypergraphs: Confidence interval and subgradient method. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 4026–4034, 2017.

[52] Hubert Chan and Zhibin Liang. Generalizing hypergraph laplacian via diffusion with mediators. In *In Proceedings of the 24th International Conference on Computing and Combinatorics (COCOON)*, pages 441–453, 2018. 17, 18

[53] Ke Tu, Peng Cui, Xiao Wang, Fei Wang, and Wenwu Zhu. Structural deep embedding for hyper-networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 426–433, 2018.

[54] Inci M Baytas, Cao Xiao, Fei Wang, Anil K. Jain, and Jiayu Zhou. Hhne: Heterogeneous hyper-network embedding. In *IEEE International Conference on Data Mining (ICDM)*, pages 875–880, 2018. 3, 18, 19

[55] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the Thirty-Third Conference on Association for the Advancement of Artificial Intelligence (AAAI)*, pages 3558–3565, 2019. 3, 4, 7, 9, 18, 24

[56] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. HyperGCN: A new method of training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, pages 1509–1520. Curran Associates, Inc., 2019. 3, 4, 7, 9, 18, 24

[57] Minyoung Choe, Sunwoo Kim, Jaemin Yoo, and Kijung Shin. Classification of edge-dependent labels of nodes in hypergraphs. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 298–309, 2023. 3

[58] Yuan Luo. Shine: Subhypergraph inductive neural network. In *Advances in Neural Information Processing Systems (NeurIPS) 35*, pages 18779–18792. Curran Associates, Inc., 2022. 6

[59] Dobrik Georgiev, Marc Brockschmidt, and Miltiadis Allamanis. HEAT: hyperedge attention networks. *Trans. Mach. Learn. Res. (TMLR)*, 2022.

[60] Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2563–2569, 2021. 4, 6, 7, 9, 24

[61] Song Bai, Feihu Zhang, and Philip H.S. Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, page 107637, 2021.

[62] Ruochi Zhang, Yuesong Zou, and Jian Ma. Hyper-{sagnn}: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations (ICLR)*, 2020. 18, 24

[63] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. Be more with less: Hypergraph attention networks for inductive text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4927–4936, 2020. 3, 6

[64] Naganand Yadati. Neural message passing for multi-relational ordered and recursive hypergraphs. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, pages 3275–3289. Curran Associates, Inc., 2020. 3

[65] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations (ICLR)*, 2022. 3, 4, 6, 7, 9, 24

[66] Peihao Wang, Shenghao Yang, Yunyu Liu, Zhangyang Wang, and Pan Li. Equivariant hypergraph diffusion neural operators. In *International Conference on Learning Representations (ICLR)*, 2023. 3, 4, 6, 7, 9, 24

[67] Jinwoo Kim, Saeyoon Oh, Sungjun Cho, and Seunghoon Hong. Equivariant hypergraph neural networks. In *The European Conference on Computer Vision (ECCV)*, pages 86–103, 2022. 3

[68] Yuxin Wang, Quan Gan, Xipeng Qiu, Xuanjing Huang, and David Wipf. From hypergraph energy functions to hypergraph neural networks. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 35605–35623, 2023. 3, 6

[69] Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzman-Saenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K. Dey, Soham Mukherjee, Shreyas N. Samaga, Neal Livesay, Robin Walters, Paul Rosen, and Michael T. Schaub. Topological deep learning: Going beyond graph data. *CoRR, arXiv:2206.00606*, 2022. 3

[70] Mathilde Papillon, Sophia Sanborn, Mustafa Hajij, and Nina Miolane. Architectures of topological deep learning: A survey on topological neural networks. *CoRR, arXiv:2304.10031*, 2023. 3

[71] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, pages 4854–4873, 2022. 4

[72] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. Heterogeneous network representation learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4861–4867, 2020.

[73] Yizhou Sun and Jiawei Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2012. 4

[74] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress?: Revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1150–1160, 2021. 4, 7, 9, 25

[75] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference (WWW)*, page 2704–2710, 2020.

[76] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference (WWW)*, page 2331–2341, 2020. 7, 9, 22, 25

[77] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 793–803, 2019.

[78] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference (WWW)*, pages 2022–2032, 2019. 4

[79] Juanhui Li, Harry Shomer, Jiayuan Ding, Yiqi Wang, Yao Ma, Neil Shah, Jiliang Tang, and Dawei Yin. Are message passing neural networks really helpful for knowledge graph completion? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pages 10696–10711, 2023. 4

[80] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[81] Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhiping Shi, Hui Xiong, and Qing He. Relational graph neural network with hierarchical attention for knowledge graph completion. In *Proceedings of the Thirty-Fourth Conference on Association for the Advancement of Artificial Intelligence (AAAI)*, pages 9612–9619, 2020.

[82] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the Thirty-Third Conference on Association for the Advancement of Artificial Intelligence (AAAI)*, pages 4424–4431, 2019.

[83] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Extended Semantic Web Conference (ESWC)*, pages 593–607, 2018. 4, 7, 9, 25

[84] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete Appl. Math.*, 1993. 4

[85] Mikko Kivelä, Alex Arenas, Marc B, James P Gleeson, Y Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014. 4

[86] Bowen Jin, Yu Zhang, Qi Zhu, and Jiawei Han. Heterformer: Transformer-based deep node representation learning on heterogeneous text-rich networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1020–1031, 2023. 9

[87] Yashu Liu, Shuang Qiu, Ping Zhang, Pinghua Gong, Fei Wang, Guoliang Xue, and Jieping Ye. Computational drug discovery with dyadic positive-unlabeled learning. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 45–53, 2017. 18

The appendix contains the following sections:

A Model Diagram and Algorithm:

We provide comprehensive visual representations of the model architecture along with a detailed algorithmic description. We elucidate the key components and their interconnections, offering a clear understanding of the structural and operational aspects of the proposed model.

B Experiments on Hyperedge Prediction:

We outline the specific experiments conducted to evaluate the model's performance in hyperedge prediction. We include details on the experimental setup, metrics used for assessment, and results obtained, contributing valuable insights into the model's efficacy in handling hypergraph structures.

C Case Study:

We present a comprehensive case study focused on a specific subject or problem. Through this case study, readers gain a deeper understanding of the real-world application and implications.

D Ablation Analysis of Attention:

Through systematic experimentation, the section provides valuable insights into the functioning of the attention mechanism, shedding light on critical aspects that influence the model's performance.

E Details and Analyses of Hyperparameters:

Here, we delve into the hyperparameters employed in the model and provide a thorough analysis of their sensitivity impact on performance.

F Dataset Construction Details:

This part elucidates the procedures and methodologies involved in constructing the dataset used for training and evaluation. It covers data sources, preprocessing steps, and any specific considerations taken to ensure the dataset relevance of the datasets.

G Details of Baselines:

We discuss the baseline models against which HEAL is compared.

H Experiments on Different Train Splits:

We explore the impact of varying training splits on the performance of the HEAL.

I Analysis of Different HFP Candidates

We discuss two candidates for feature propagation.

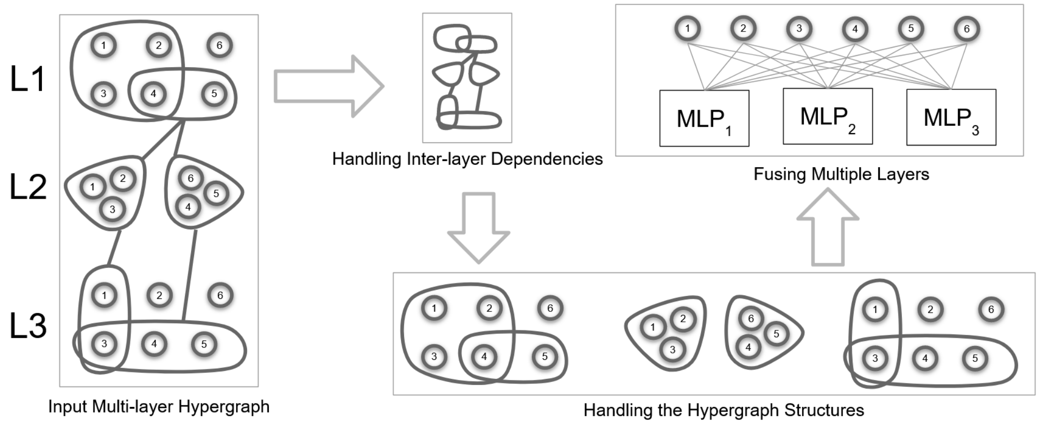


Figure 3: Visual Representation of the Three Stages in the HEAL Method. The diagram illustrates a three-layer hypergraph with six vertices. In the initial stage, inter-layer dependencies are handled using a graph feature propagator. Subsequently, the hyperedges within each layer are handled through a hypergraph feature propagator. Finally, a fusion of information across layers is achieved using Multi-Layer Perceptrons (MLPs) equipped with a common hidden space for each vertex.

A Model Diagram and Algorithm

Algorithm 1 HEAL

Require: $\mathcal{H} = (V, E, X, L)$: Input Multi-layer Hypergraph, a dataset-specific parameter τ
Ensure: $H_v,\ \forall v \in V$: Hidden vertex representations

function GFP($\mathcal{H}$) ▷ Handling Inter-layer Dependencies
  Initialise attribute matrix $X_e \in \mathbb{R}^{m \times |L|}$ ▷ $m = |E|$ is the number of directed hyperedges
  for $(t, h, l, \ell) \in E$ **do**
    Assign one-hot encoding of l to X_t and that of ℓ to X_h
  $\tilde{X}_e = \tilde{A}^2 X_e$ ▷ $\tilde{A}$ is the normalised adjacency with self-loops
  return $\tilde{X}_e$
function HFP($I^{(l)}, X_v^{(l)}, \tilde{X}_e$) ▷ Handling the Hypergraph Structures
  Obtain zero padded $\bar{X}_v^{(l)}$ by appending zero vectors of size $|L|$ to row vectors in $X_v^{(l)}$
  Obtain zero padded $\bar{X}_e^{(l)}$ by appending $\tilde{X}_e$ to zero vectors of size d_l
  for $u \in V$ **do**
    Initialise $\tilde{X}_v^{(l)} = \bar{X}_v^{(l)}$
    Initialise $\tilde{X}_e^{(l)} = \bar{X}_e^{(l)}$
    while $||\tilde{X}_{v,u}^{(l)} - \bar{X}_{v,u}^{(l)}||_2 > \tau$ **do**
      $\tilde{X}_v^{(l)} = D_v^{-1} I \tilde{X}_e^{(l)}$
      $\tilde{X}_e^{(l)} = D_e^{-1} I^T \tilde{X}_v^{(l)}$
    Let $\chi_u^{(l)} = \tilde{X}_{v,u}^{(l)}$
  return $\tilde{X}_v^{(l)}$ consisting of $\chi_u^{(l)}$ as row vectors for $u \in V$
function FUSION($\tilde{X}_v^l,\ v \in V,\ l \in L$) ▷ Fusing Multiple Layers
  for $(v, l) \in V \times L$ **do**
    $H_v^{(l)} = MLP_l(\tilde{X}_v^{(l)})$
  for $v \in V$ **do**
    $H_v = \text{Attention}(\{H_v^l\}_{l \in L})$
  return $H_v, v \in V$

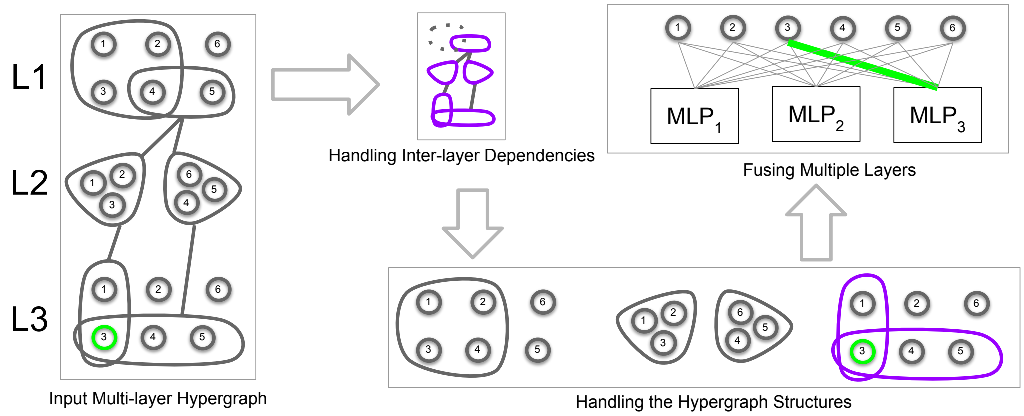


Figure 4: (Best seen in colour) Visual representation of the three stages in the HEAL method from the perspective of a vertex at a layer. Consider vertex 3 (coloured green) at layer 3 (i.e., L3). (1) In the first stage (Handling Inter-layer Dependencies), hyperedges at layer Li are represented by one-hot encodings (of size 3 in this example) with a one at the ith position and zeroes elsewhere. Then the graph feature propagator (GFP) propagates the encodings along the edges given by inter-layer dependencies. In the figure hyperedges that propagate among them are coloured in purple. The one hyperedge that does not participate is dotted. (2) In the second stage, a hypergraph feature propagator (HFP) is used to propagate features from hyperedges to vertices and then to co-vertices (i.e., those in the same hyperedge). Through this process, the vertex 3 receives features of layers and other vertices via its incident hyperedges. From the perspective of vertex 3 coloured green, it receives propagated from its two incident hyperedges coloured purple. (3) In the third stage, we use an MLP from each layer followed by attention to fuse features from all layers. In the example, the green coloured link represents the fact that features from vertex 3 are fed as input to an attention layer to compute the final vertex representation of vertex 3.

B Experiments on Hyperedge Prediction

In order to assess the effectiveness of our proposed HEAL for hyperedge prediction, we conducted a series of experiments on real-world networks.

Real-world Example. Let us take the scenario of predicting hyperedges within academic networks as an illustration (for instance, predicting author collaborations). In this context, we can conceptualise each year of document publication as a distinct layer. Here, authors are depicted as vertices, documents as hyperedges, and citations between documents as inter-layer connections. To illustrate this, please see a toy example provided in Figure 5.

B.1 Notation

Let $E_U := \Big\{(t,l) : (t,h,l,\ell) \in E\Big\} \cup \Big\{(h,\ell) : (t,h,l,\ell) \in E\Big\}$.

The set of hyperedges at layer l is

$$E_U^l := \Big\{e : (e,l) \in E_U\Big\}. \tag{2}$$

B.2 Interaction function

We discuss how higher-order composite interactions in $E_U^l,\ l \in L$, in Equation 2, are preserved in the embedding space without a limit on the hyperedge size. We design a novel interaction function, $I_e : 2^V \to [0,1]$, which given $e \subseteq V$, takes the form $I_e = \sigma\Big(W \cdot g\Big(\big\{z_v\big\}_{v \in e}\Big) + b\Big)$, W is a parameter of dimension $1 \times r$, σ is the sigmoid function. The score, I_e, of e, is to be higher than that of any vertex set that is not a hyperedge in $\mathcal{H}$. Inspired by non-linear Laplacians [52]:

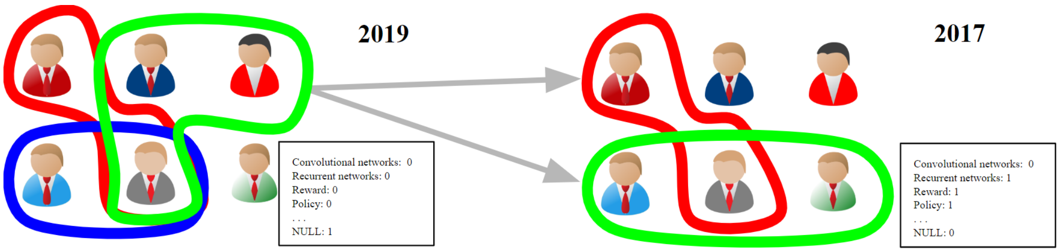


Figure 5: (Best seen in colour) An example of an attributed multi-layer hypergraph. Vertices represent authors (6 here). Layers represent years (2019 and 2017). Colours are used to just distinguish different hyperedges in a layer. Intra-layer hyperedges represent documents (3 in 2019 and 2 in 2017). Inter-layer directed hyperedges represent citations (2 citations here). Attributes represent author interests (bag of word features). Attributes are shown for one author (green coloured) in 2019 and 2017.

Table 5: Results of baselines and HEAL.

	DBLP-HP			Twitter			Amazon		
Model	ROC-AUC	Avg. P	P@K	ROC-AUC	Avg. P	P@K	ROC-AUC	Avg. P	P@K
GCN [14]	55.36	62.27	57.85	58.61	57.90	57.73	70.04	69.57	69.81
GAT [15]	55.67	62.24	57.74	58.84	57.93	58.01	70.12	69.54	69.88
HGNN [55]	57.88	63.36	58.17	67.25	67.08	64.74	74.02	74.95	74.26
HyperGCN [56]	57.93	63.25	58.19	67.14	67.32	64.89	73.99	74.52	74.38
HHNE [54]	61.24	70.19	59.76	69.11	69.02	66.78	75.13	76.72	75.02
Hyper-SAGNN [62]	61.04	70.03	59.83	70.35	69.47	67.14	75.01	76.96	75.21
DMGI [41]	58.57	64.92	58.14	67.98	68.97	66.57	74.05	73.38	73.98
HDMI [42]	60.45	67.96	60.03	68.59	68.93	66.47	74.86	74.72	74.33
RGCN [62]	60.87	69.44	59.37	69.96	69.54	66.86	74.32	75.98	74.45
SHGN [62]	60.96	70.08	59.39	70.18	69.29	66.8	74.57	76.11	75.04
HEAL (proposed)	**66.36**	**73.71**	**64.19**	**71.72**	**70.43**	**68.19**	**76.48**	**78.57**	**75.34**

$$I_e := \sigma\left(\frac{1}{|e|} W \cdot \mathrm{med}\Big\{z_v\Big\}_{v\in e} + b\right). \tag{3}$$

where, given a set of r-dimensional vectors $z_1, \cdots, z_m$,

$$\mathrm{med}\Big\{z_1, \cdots, z_m\Big\} = \left(\Big(\max_{s\in[m]} z_{sk} - \min_{i\in[k]} z_{ik}\Big)^2 + \sum_{j=1}^{m}\Big(\max_{s\in[m]} z_{sk} - z_{jk}\Big)^2 + \sum_{j=1}^{m}\Big(z_{jk} - \min_{i\in[m]} z_{ik}\Big)^2\right)_{k=1,\cdots,r}.$$

here $k = 1, \cdots, r$ is an index used to represent each of the r components in the vectors. s denotes supremum, i infimum, and med the mediator Laplacian [52].

B.3 Objective function

In our work, unknown interactions denoted by $F_U^l, l \in L$ are sampled from $2^V - E_U^l,\ l \in L$. The loss function is based on ranking [87]:

$$\mathcal{L} = \sum_{l=1}^{|L|} \frac{1}{|E_U^l|} \sum_{e\in E_U^l} \Lambda\left(\frac{1}{|F_U^l|} \sum_{\bar{e}\in F_U^l} I_{\bar{e}} - I_e\right). \tag{4}$$

$\Lambda(x)$ is a non-decreasing function such as the logistic function $\Lambda(x) = \log(1 + e^x)$. $\mathcal{L}$ maximises the number of hyperedge scores higher than the average score of the sampled sets (we ensure the sizes are the same, i.e., $|F_U^l| = |E_U^l|$ in experiments).

B.4 Training Details

To sample unknown sets of vertices from $2^V - E_U^l$, we use the following: For each hyperedge $e \in E_U^l,\ l \in L$, we create a corresponding $\bar{e} \in 2^V - E_U^l$ by having half of the vertices, i.e., $\frac{|e|}{2}$ sampled from e, the remaining half from $V - e$ and then adding $\bar{e}$ to F_U^l. To avoid bias in the sizes, we ensure that each hyperedge e has a "corresponding" vertex set $\bar{e}$ of the same size.

We hide 20% of the known hyperedges (equally distributed in all layers) during training and use 80% of the known hyperedges and sampled non-interacting vertex sets to train all the models. Of the 20% hidden hyperedges, we use 10% for model selection for all methods. The remaining 10% of the data are held out and used for testing.

B.5 Metrics for Evaluation

We use commonly used evaluation criteria, i.e., the area under ROC curve (ROC-AUC), Average precision (Avg. P), and Precision@K (P@K) where K is the number of missing (test) hyperedges. Test interaction scores were sorted in decreasing order to compute Avg.P and P@K.

Table 6: Comparison on DBLP-HP, Amazon, and Twitter over 10 random splits.

	DBLP-HP			Amazon			Twitter		
Model	ROC-AUC	Avg. P	P@K	ROC-AUC	Avg. P	P@K	ROC-AUC	Avg. P	P@K
HDMI [42]	60.52 ± 0.6	68.01 ± 0.3	59.99 ± 0.4	68.67 ± 0.4	68.57 ± 0.5	66.68 ± 0.5	74.32 ± 0.5	74.95 ± 0.6	74.19 ± 0.3
HHNE [54]	60.97 ± 0.5	70.04 ± 0.6	59.64 ± 0.2	69.03 ± 0.6	68.93 ± 0.3	66.53 ± 0.4	74.77 ± 0.4	76.86 ± 0.6	74.93 ± 0.3
HEAL	$\mathbf{65.92 \pm 0.6}$	$\mathbf{73.63 \pm 0.5}$	$\mathbf{64.26 \pm 0.6}$	$\mathbf{71.34 \pm 0.5}$	$\mathbf{70.53 \pm 0.2}$	$\mathbf{68.32 \pm 0.4}$	$\mathbf{76.32 \pm 0.3}$	$\mathbf{78.47 \pm 0.2}$	$\mathbf{75.30 \pm 0.5}$

B.6 Results and Discussion

The results of hyperedge prediction experiments are shown in Table 5. In a general sense, our suggested HEAL consistently outperforms all baseline methods across commonly used metrics for hyperedge prediction. The superior performance of HEAL can be ascribed to its proficiency in capturing advanced higher-order interactions through the utilisation of both the hypergraph feature propagator module and the innovative interaction function proposed in this study.

Table 6 shows performance of HEAL and the two most competitive baselines for 10 random train-test splits. HEAL produces statistically significant outcomes, as confirmed by the Welch t-test. The highest p-value observed across all datasets remains under 0.001.

C Case Study

We investigated a case study of HEAL on the IMDB movie dataset.

In the dataset the following are the components of the multi-layer hypergraph:

- Vertices: Movies
- Vertex features: Movie plot
- Hyperedges: Common cast member
- Layers: Actors, Directors
- Cross-layer Interactions: A cast member (e.g., Tom Hanks) who has acted in some movies and directed in some movies

HEAL utilises cross-layer interactions by propagating layer information across hyperedges. This propagated information is then merged with movie plot details and further disseminated to movie vertices. By leveraging the comprehensive information within the multi-layer hypergraph, HEAL precisely matches movie genres, as demonstrated with movies like 'Road To Perdition,' where Tom Hanks serves as an actor

Ground-truth: 'Drama', 'Thriller', 'Crime' HEAL: 'Drama', 'Thriller', 'Crime' Best that any Baseline could do: 'Drama', 'Thriller', 'Action', 'Crime'

D Ablation Study of Attention

In this section we conduct an ablation of the the fusion across different layers which is one of the key components of HEAL. Table 7 and Table 8 show the comparisons of HEAL (with attention layer) against mean pooling and max pooling baselines in which the attention layer was removed.

Table 7: Ablation Study of Attention on Hyperedge Prediction.

Model	DBLP-HP ROC-AUC	Avg. P	P@K
Mean Pooling	60.31 ± 0.5	69.55 ± 0.6	59.27 ± 0.2
Max Pooling	61.37 ± 0.4	71.01 ± 0.5	60.12 ± 0.5
HEAL	$\mathbf{65.92 \pm 0.6}$	$\mathbf{73.63 \pm 0.5}$	$\mathbf{64.26 \pm 0.6}$

Table 8: Ablation study of attention layer on the IMDB node classification datasets.

Method	IMDB-MC		IMDB-ML	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Mean Pooling	68.11±1.07	68.26±0.68	51.95±1.18	55.42±0.90
Max Pooling	67.91±0.85	68.04±0.36	52.08±1.19	56.29±0.65
HEAL	**69.14±0.87**	**69.46±0.54**	**53.59±1.18**	**57.88±0.62**

E Details and Analyses of Hyperparameters

In this section, we describe the hyperparameter settings and perform sensitivity analyses of key hyperparameters.

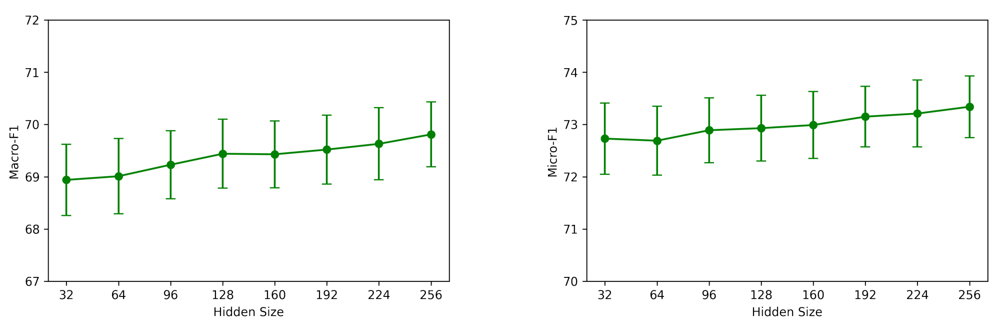
E.1 Hyperparameter Settings

The hidden representation size of MLPs in the proposed fusion component is selected through grid search in the range $\{32, 64, 128, 256\}$. We use the Adam optimiser with a learning rate of 0.001 to train for a maximum of 1000 epochs. The dropout rate is 0.5. The threshold for the number of smoothing iterations of HFP, i.e., τ, is selected in the range $\tau \in \{0.05, 0.075, 0.1, 0.125, 0.15\}$.

E.2 Sensitivity Analysis

We study the effect of varying the hidden size and threshold τ

Hidden Size. The effect of varying hidden sizes on HEAL's performance is depicted in Figure 6.



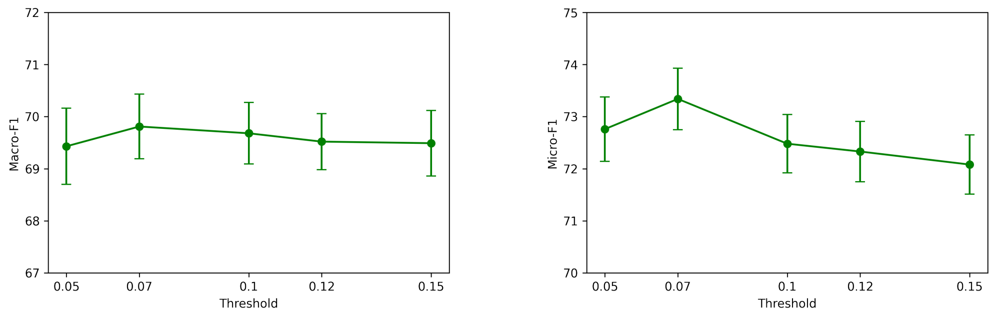
(a) Macro-F1 **(b)** Micro-F1

Figure 6: Hidden size sensitivity. Macro-F1 and Micro-F1 scores on the CoraLCA dataset.

The findings suggest that HEAL is capable of capturing and leveraging meaningful information from the graph structure across a range of hidden sizes. This flexibility in accommodating different hidden sizes enhances the adaptability and robustness of HEAL to changes in hidden sizes.

E.3 Analyses of Threshold

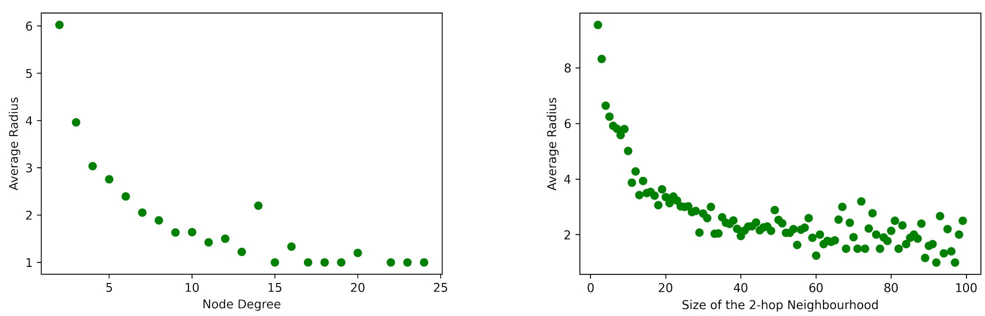
Sensitivity Analysis. The effect of varying embedding sizes on HEAL is depicted in Figure 7. The findings suggest that HEAL is capable of capturing and leveraging meaningful information from the hypergraph structure across a range of threshold values. The choice of an optimal threshold value is essential for improving the resilience of models.



(a) Macro-F1 **(b)** Micro-F1

Figure 7: Threshold sensitivity. Macro-F1 and Micro-F1 scores on the CoraLCA dataset.

Smoothing Iterations for a given Threshold as a Function of Vertices. In this paragraph, we delve into the threshold τ used to control the number of smoothing iterations. In Figure 8a, we examine the relationship between the vertex degree and the average smoothing iterations, averaged across all nodes with a particular degree. Figure 8b delves into the interplay between the number of smoothing iterations and the size of the 2-hop neighbourhood. The number of vertices in the 2-hope neighbourhood offers insights into the density of connectivity in the vicinity of each node. The observations corroborate our intuition, indicating that well-connected vertices typically require smaller number of iterations, while vertices with limited connectivity benefit from a greater number of hops of smoothing iterations.



(a) 1-hop neighbourhoods **(b)** 2-hop neighbourhoods

Figure 8: Smoothing Iterations as a Function of Vertex Connectivity

E.4 Training Time, Test Performance Tradeoff

In this section, we explore the relationship between training time and Macro-F1, examining the tradeoff between the two factors. The findings in Figure 9, providing insights into the relationship between training time and test performance. The proposed smoothing strategy, generally is faster and achieves impressive F1 scores.

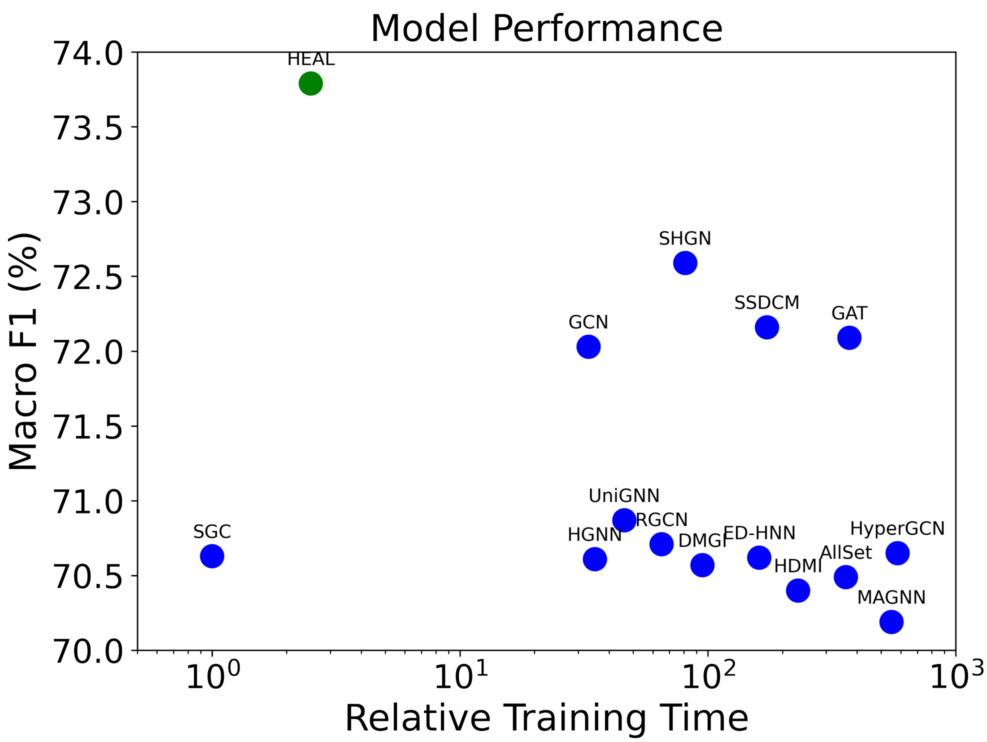


Figure 9: Visualising the relative training times and Macro-F1 tradeoff of the proposed method (green) and baselines (blue) on DBLP-LCA.

F Dataset Construction Details

Our experimentation covered four node classification datasets and three hyperedge prediction datasets. Details pertaining to the construction of each of the seven datasets are elaborated upon in this section.

F.1 Movie Datasets: IMDB-MC and IMDB-ML

In this paper, IMDB-MC is a multi-class dataset and IMDB-ML is a multi-label dataset. We assembled the IMDB dataset by following a step-by-step procedure as follows:

- Download the data from the github code page [4] of a prior research work [76]
- Depict the movies as vertices and utilise movie titles and plot keywords to construct vertex features using the bag-of-words representation.
- Build multi-layer hyperedges using movie cast data. This involves organizing actors into one layer, directors into another, so on, and establishing inter-layer connections for cast members with multiple roles across different movies.
- Represent movie genres as output vertex labels in the context of multi-label classification (IMDB-ML) and output vertex classes for multi-class classification (IMDB-MC).

IMDB-MC involves forming a 4-class dataset by selecting movies belonging to 'Drama', 'Action', 'Comedy', and 'Others' genres, which collectively cover all other movie genres. In the context of IMDB-ML, we keep all the movie genres.

F.2 Lead Co-Authorship Datasets: CoraLCA and DBLP-LCA

We employed similar approaches to construct the lead co-authorship datasets for vertex classification.

[4] https://raw.githubusercontent.com/cynricfu/MAGNN/master/data/raw/IMDB/movie_metadata.csv

- Download the data from the sources of Cora [5] and DBLP [6]
- Depict the documents as vertices and utilise document titles and document abstracts to construct vertex features using the bag-of-words representation.
- Build multi-layer hyperedges using authorship data. This involves organising lead authors into one layer, non-lead co-authors into another, and establishing inter-layer connections for authors with lead authors in some documents and co-authors in others across different documents.
- Represent document categories as output vertex classes for multi-class classification.

Dataset	#Nodes	#Features	# Layer 1	# Layer 2	Inter-layer	Task
IMDB	5425	8851	6256	163	2399	5 (MC), 29 (ML)
CoraLCA	2708	1433	1659	1037	578	7-class classification
DBLP-LCA	43891	1092	32589	22981	984	6-class classification

Table 9: Statistics of the node classification datasets

F.3 DBLP-HP

This dataset is a hyperedge prediction (HP) dataset. THe difference between this dataset and the node classification dataset lies in how the mutli-layer hypergraph is constructed. The details are as follows:

- Extract the data from aminer and consider documents from AI, ML, Data Mining Conferences
- Depict the authors as vertices and utilise document titles and document abstracts to construct vertex features using the bag-of-words representation. We take the average bag-of-word features of all the documents co-authored by an author to construct author features (i.e., vertex features)
- Build a multi-layer hypergraph with years of publication as layers, colloborations (i.e., documents) as hyperedges, and citations between documents as inter-layer connections. Figure 5 shows a toy example

We use 5% of each vertex classification dataset for training, 25% for model selection, and the remaining 70% for testing all the methods.

F.4 Twitter

This dataset [7] is about tweets related to the discovery of Higgs boson between 1st and 7th, July 2012. It is made up of four directional relationships betwen more than 450, 000 Twitter users. The layers are re-tweet, mention, and friendship / follower between Twitter users.

F.5 Amazon

This is a 2-layer hypergraph of co-purchase and co-view relationships [8]. Items represent vertices and items purchased together are layer one hyperedges and items viewed together are layer two hyperedges. Items purchased and viewed by the same customer form inter-layer connections. We use only the product metadata of Electronics category.The attribtes include the price, sales- rank, brand, and category.

Table 10: Layer-wise statistics of DBLP-HP dataset. Number of features used by baselines is 5395

layer number	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
# hyperedges (collaborations)	1393	2443	2472	3508	4043	4763	4823	5396	6958	1703
# features (bag-of-word)	947	1585	1555	1986	2260	2499	2525	2795	3240	1324

[5] https://people.cs.umass.edu/ mccallum/data.html
[6] https://www.aminer.org/DBLP_Citation
[7] https://snap.stanford.edu/data/higgs-twitter.html
[8] http://jmcauley.ucsd.edu/data/amazon/links.html

G Details of Baselines

In this section, we provide an overview of the baselines, categorised by their types, which are used for the purpose of comparison.

Graph Models are executed on an input graph, where edges are established between two vertices if they co-appear in a hyperedge on any layer. The edge weight corresponds to the number of times the two vertices co-appear across hyperedges and layers. Cross-layer hyperedges are not integrated into these methods, as it is not obvious how to effectively combine them with pairwise interactions.

- **GCN** [14] originated as a highly efficient convolutional approach for semi-supervised classification on graph-structured data. Its effectiveness and adaptability have led to its widespread use across various domains.
- **SGC** [33] simplifies the process of graph-based learning by discarding non-linearities in GCN and merging weight matrices between consecutive layers.
- **GAT** [15] employs softmax attention layers to grant specific weights to nodes within a neighborhood, facilitating enhanced learning of node representations.

Hypergraph Models operate on an input hypergraph where the set of hyperedges is formed by taking the union of all hyperedges existing in all layers. The current state-of-the-art in this research domain relies on different techniques (e.g., attention) on *the star expansion of hypergraphs*. To include cross-layer hyperedges in these methods, we add an edge between two hyperedge vertices in the star expansion if there is a cross-layer interaction between the two hyperedges.

- **HGNN** [55] technique involves enlarging the input hypergraph through clique expansion and then utilising GCN on this expanded hypergraph
- **HyperGCN** [56] employs non-linear Laplacians to facilitate the propagation of embeddings along pairs of nodes within hyperedges. We choose the top-performing model, which is the one incorporating mediators in the non-linear Laplacian.
- **UniGNN** [60] puts forth permutation-invariant functions for aggregating messages from vertices and hyperedges. We select UniGCN due to its superior performance on our datasets.
- **AllSet** [65] involves the composition of two multiset functions: one that maps vertices to hyperedges and another that performs the reverse mapping. AllSetTransformer stands out as our selection because of its outstanding performance on our datasets.
- **ED-HNN** [66] is a recent equivariant-based approach on the star expansion.
- **Hyper-SAGNN** [62] is a self-attention-based approach on hypergraphs designed for hyperedge prediction.

Multi-layer Graph Models process an input multi-layer graph, maintaining the layer structure while converting all hyperedges within each layer into pairwise edges. Within each layer, the edge weight is determined by the frequency of co-occurrence between two vertices across hyperedges in that specific layer.

The baselines propose a regularisation term that can easily be extended to accommodate pairwise cross-layer dependencies, represented as cross-layer edges. The assumed pairwise dependency is that a vertex in one layer is connected to a cross-layer vertex if the former appears in a cross-layer hyperedge with the latter.

- **DMGI** [41] introduces an approach that maximizes mutual information and employs a consensus regularization framework to reduce disparities among node embeddings specific to different relation types.
- **HDMI** [42] broadens the scope of mutual information maximisation to encompass mutual dependence between node embeddings and node attributes. Additionally, it introduces an attention-based fusion module designed to merge node embeddings from various layers of the multiplex network.
- **SSDCM** [43] combines local patch representations at the node level with globally structured graph representations that are correlated with labels. This strategy is designed to effectively model node and cluster representations throughout the layers of a multiplex network.

Heterogeneous Graph Models work with a heterogeneous graph representation where vertices and layered hyperedges are all represented using typed vertices and typed edges. Cross-layer dependencies, which are unexplored in the literature, are modelled by additional typed edges connecting two hyperedge vertices if they are involved in a cross-layer interaction.

- **RGCN** [83] expands upon the Graph Convolutional Network (GCN) to accommodate relational graphs with multiple edge types. The convolution process in RGCN can be understood as a weighted combination of standard graph convolutions applied to distinct edge types.
- **MAGNN**[76] introduces meta-path encoders that capture information along the entire meta path
- **SHGN**[74] proposes learnable edge-type embedding with residual connections for embedding heterogeneous networks.

H Experiments on Different Train Splits

To gauge how HEAL responds to fluctuations in training set sizes, we modified the sizes of the training set within the IMDB-ML dataset. The results are shown in Figure 11.

Table 11: F-1 scores on held-out test sets of representative baselines and our approach on the IMDB multi-label movie genre prediction task.

	5%		10%		20%		30%	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
MLP	46.56	50.08	48.67	51.96	51.18	55.92	54.04	57.89
NDLS	49.03	51.57	53.28	56.90	55.73	59.02	56.96	60.13
AllSet	48.95	53.28	53.99	57.17	56.15	59.97	57.36	61.09
SSDCM	52.19	55.17	53.91	57.21	56.04	59.87	57.26	61.22
SHGN	51.59	55.77	53.39	57.14	56.13	59.99	57.44	61.63
HEAL	**53.54**	**57.95**	**54.55**	**58.34**	**56.64**	**60.53**	**58.25**	**62.74**

I Analysis of Different HFP Candidates

Table 12: Performance Comparison of Candidates for HFP in HEAL.

Hypergraph Feature Propagator (HFP) in HEAL	IMDB-MC		IMDB-ML	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
SGC	68.19±1.08	68.33±0.74	51.93±1.04	55.42±0.86
S^2GC	68.24±0.98	68.42±0.47	52.01±1.05	55.76±0.63
Proposed	**69.14±0.87**	**69.46±0.54**	**53.59±1.18**	**57.88±0.62**

Table 12 shows the performance of potential candidates considered for the hypergraph feature propagator (HFP) module within HEAL. The introduced HFP effectively tackles the balance between undersmoothing due to insufficient baseline layers and oversmoothing resulting from excessive baseline layers. This is achieved by adapting to varying numbers of smoothing iterations for nodes.