HDMI: High-order Deep Multiplex Infomax

Baoyu Jing
baoyuj2@illinois.edu
University of Illinois at Urbana-Champaign
IL, USA

Chanyoung Park
cy.park@kaist.ac.kr
Department of Industrial and Systems Engineering, KAIST
Daejeon, Republic of Korea

Hanghang Tong
htong@illinois.edu
University of Illinois at Urbana-Champaign
IL, USA

ABSTRACT

Networks have been widely used to represent the relations between objects such as academic networks and social networks, and learning embedding for networks has thus garnered plenty of research attention. Self-supervised network representation learning aims at extracting node embedding without external supervision. Recently, maximizing the mutual information between the local node embedding and the global summary (e.g. Deep Graph Infomax, or DGI for short) has shown promising results on many downstream tasks such as node classification. However, there are two major limitations of DGI. Firstly, DGI merely considers the *extrinsic* supervision signal (i.e., the mutual information between node embedding and global summary) while ignores the *intrinsic* signal (i.e., the mutual dependence between node embedding and node attributes). Secondly, nodes in a real-world network are usually connected by multiple edges with different relations, while DGI does not fully explore the various relations among nodes. To address the above-mentioned problems, we propose a novel framework, called High-order Deep Multiplex Infomax (HDMI), for learning node embedding on multiplex networks in a self-supervised way. To be more specific, we first design a joint supervision signal containing both extrinsic and intrinsic mutual information by high-order mutual information, and we propose a High-order Deep Infomax (HDI) to optimize the proposed supervision signal. Then we propose an attention based fusion module to combine node embedding from different layers of the multiplex network. Finally, we evaluate the proposed HDMI on various downstream tasks such as unsupervised clustering and supervised classification. The experimental results show that HDMI achieves state-of-the-art performance on these tasks.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Learning latent representations*; • **Mathematics of computing** → **Information theory**; **Graph algorithms**.

KEYWORDS

Network Representation Learning, Multiplex Networks, High-order Mutual Information

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
WWW '21, April 19–23, 2021, Ljubljana, Slovenia
© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.
ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449971

ACM Reference Format:
Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. HDMI: High-order Deep Multiplex Infomax. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3442381.3449971

1 INTRODUCTION

Network or graph structures have been widely used to represent the relations among objects, such as the citation relation among papers, the co-author relation among researchers, as well as the friendship relation among people. Mining and discovering useful knowledge from networks has been an active research area, within which network representation learning has garnered substantial research attention and it has been demonstrated to be effective for various tasks on networks [6, 45, 48], such as node classification, node clustering and link prediction.

Self-supervised network representation learning aims at extracting node embedding without introducing external supervision signals. The dominant strategy of the self-supervised network representation learning is to design a signal based on the *proximity* of the nodes in the network, such that the node embedding will retain the proximity. For example, given a node, DeepWalk [30] and node2vec [9] maximize the probabilities of its neighbors sampled by random walks. LINE [34] maximizes the probabilities between a node and its first or second-order proximate neighbors. Albeit the effectiveness of these methods, the proximity based supervision signals only capture the *local* characteristic of networks. With the introduction of the powerful network encoders, such as Graph Convolutional Network (GCN) [16] which can naturally capture such local proximity based on graph convolution [3], the improvement brought by the traditional proximity based supervision signals is limited [37]. To further improve the quality of node embedding, Deep Graph Infomax (DGI) [37] uses GCN as the network encoder and trains it by maximizing the mutual information between the node embedding and the *global* summary of the network. However, there are two major limitations of DGI.

One limitation of DGI is that it focuses on the *extrinsic* supervision signal (i.e., whether the node embedding and the global summary come from the same network), and it does not fully explore the *intrinsic* signal (i.e., the mutual dependence between the embedding vector and the attribute vector of a node). Node attributes themselves often contain discriminative information about the nodes, and capturing the mutual dependence between them helps the embedding obtain more discriminative information and thus discover the potential relations among nodes. For example, in the citation network, papers always contain textual attributes such as abstracts and keywords. Two papers focusing on different sub-areas (e.g. dynamic graph algorithms and sub-graph mining)

might not have a direct citation relation, but they might share the same or similar keywords (e.g. social network). In such a case, they are likely to belong to the same area, e.g., social analysis or web mining. The extrinsic signal only models the global property of the citation network and ignores the intrinsic mutual dependence, and thus it might be unable to discover the hidden relation between the two papers. The existing dominant strategy for capturing the mutual dependence between embedding and attributes is using reconstruction error of attributes given node embedding [8, 24, 46]. However, as pointed out by [41], there is no substantial connection between the discriminative ability (or quality) of embedding with the reconstruction loss. In this paper, we propose to use mutual information between node embedding and node attributes as the intrinsic supervision signal. Moreover, to further capture the synergy between the extrinsic and the intrinsic signals, we propose to use high-order mutual information among the node embedding, global summary vector, and node attributes, and we propose a novel High-order Deep Infomax (HDI) to maximize the high-order mutual information.

Another limitation of DGI is that it assumes a single type of relation among the nodes in a network, but nodes in a network are usually connected via multiple edges with different relation types. For example, in an academic network, two papers can be connected via shared authors or citation relation. In a product network, products can be linked by relations such as AlsoView, AlsoBought, and BoughtTogether. In a social network, two people can be connected by many relations such as direct friendship, the same school, and the same employer. A network with multiple types of relations is referred to as a multiplex network, and a multiplex network can be decomposed into multiple layers of networks, where each layer only has one type of relation. Figure 1 provides an example of a multiplex network. The simplest way to learn node embedding for a multiplex network is to first extract node embedding independently from different layers and then use average pooling over the node embedding from different layers to obtain final embedding for each node. However, as observed by many recent studies [4, 28, 33, 39, 44], layers are related with each other and they can mutually help each other for downstream tasks. For example, in the academic multiplex network mentioned above, the citation network layer and the shared author network layer usually provide different aspects of the papers' subject. To be more specific, the citation layer usually links a paper (e.g. about the attributed network) with related papers from other areas (e.g. computer vision), while a specific author usually focuses on a specific area (e.g. network representation learning). To capture such a mutual relation, the attention mechanism [1] is mostly adopted to calculate the weights for different layers [31, 39]. However, these methods require external supervision (e.g. labels of nodes) to train the attention module. Recently, DMGI [28] proposes a complex combination module, which first uses an attention mechanism to obtain reference node embedding and then uses a consensus regularization to obtain final node embedding. In this paper, we propose an alternative semantic attention [43] based method as the fusion module to combine node embedding from different layers. More importantly, different from existing works, the proposed fusion module is trained via the proposed high-order mutual information.

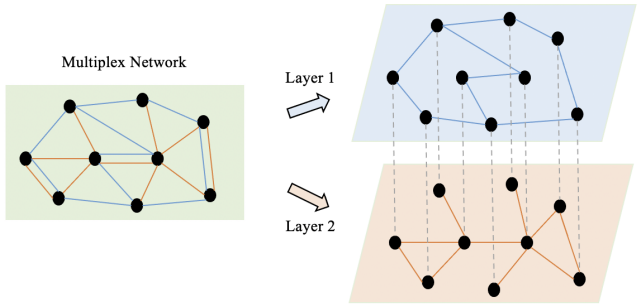


Figure 1: An example of the multiplex network, where different colors represent different types of relations among nodes. A multiplex network contains multiple layers of networks, where each layer has one type of relations.

Our main contributions are summarized as follows:

- We propose a novel supervision signal based on the high-order mutual information, which combines extrinsic and intrinsic mutual information, for learning network embedding on both attributed networks and attributed multiplex networks.
- We introduce a novel High-order Deep Infomax (HDI) to optimize the proposed high-order mutual information based supervision signal.
- We propose an attention based fusion module to combine node embedding from different layers of a multiplex network, which is trained via the high-order mutual information based supervision signal.
- We evaluate the proposed methods on a variety of real-world datasets with various evaluation metrics to demonstrate the effectiveness of the proposed methods.

The rest of the paper is organized as follows. We briefly introduce the attributed multiplex network, mutual information and DGI in Section 2. We introduce the high-order mutual information, as well as the proposed HDI and HDMI in Section 3. The experimental results are presented in Section 4. We provide a brief review of the most relevant works in Section 5. Finally, we conclude the paper in Section 6.

2 PRELIMINARIES

In this section, we first introduce relevant concepts for attributed multiplex networks. Then we introduce mutual information related concepts as well as Deep Graph Infomax (DGI) [37]. We also summarize the notations used in this paper in Table 1.

2.1 Attributed Multiplex Networks

An attributed multiplex network (Definition 2.2) is comprised of multiple layers of attributed networks (Definition 2.1).

Definition 2.1 (Attributed Network). An attributed network is represented by $\mathcal{G}(A, F)$, where $A \in \mathbb{R}^{N \times N}$ denotes adjacency matrix and $F \in \mathbb{R}^{N \times d_F}$ denotes the attribute matrix, N and d_F denote the number of nodes and the dimension of attributes respectively.

Definition 2.2 (Attributed Multiplex Network). An attributed multiplex network $\mathcal{G}_{\mathcal{M}} = \{\mathcal{G}^1, \cdots, \mathcal{G}^R\}$ is comprised of $R \geq 1$ layers of attributed networks, where $\mathcal{G}^r(A^r, F)$ ($r \in [1, \cdots, R]$, $A^r \in \mathbb{R}^{N \times N}$, $F \in \mathbb{R}^{N \times d_F}$) denotes the r-th layer. Note that different layers capture different types of relations between nodes, and all of the layers share the same node attribute matrix.

2.2 Mutual Information

Mutual information measures the mutual dependence between two random variables, which is based on Shannon entropy, and its formal definition is given in Definition 2.3. In order to measure the mutual dependence between multiple random variables, the concept of high-order/multivariate mutual information is introduced in the information theory [23].

Definition 2.3 (Mutual Information). Given two random variables X and Y, the mutual information between them is defined by:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y) \quad (1)$$

where $H(X)$ and $H(X|Y)$ denote entropy and conditional entropy respectively, $H(X,Y)$ denotes the entropy for the joint distribution of X and Y.

Definition 2.4 (High-order Mutual Information [23]). High-order mutual information is a generalization of mutual information on $N \geq 3$ random variables. Given a set of N random variables $X_1, \cdots, X_N$, the high-order mutual information is defined analogous to the definition of mutual information (Definition 2.3):

$$I(X_1; \cdots; X_N) = \sum_{n=1}^{N} (-1)^{n+1} \sum_{i_1 < \cdots < i_n} H(X_{i_1}, \cdots, X_{i_n}) \quad (2)$$

where $H(X_{i_1}, \cdots, X_{i_n})$ denotes the joint entropy for $X_{i_1}, \cdots, X_{i_n}$, and the summation $\sum_{i_1 < \cdots < i_n} H(X_{i_1}, \cdots, X_{i_n})$ runs over all of the combinations of random variables ($\{i_1, \cdots, i_n\} \in [1, \cdots, N]$).

High-order mutual information not only captures the mutual information between each pair of two random variables but also the synergy among multiple random variables. Equation (4)-(6) provide an example when $N = 3$.

2.3 Deep Graph Infomax

DGI is a self-supervised learning or an unsupervised learning method for learning node embedding on attributed networks (Definition 2.1), the main idea behind which is to maximize the mutual information between node embedding $\mathbf{h}$ and the global summary vector $\mathbf{s}$ of the entire network $\mathcal{G}$: $I(\mathbf{h}, \mathbf{s})$.

When maximizing $I(\mathbf{h}, \mathbf{s})$, DGI leverages negative sampling strategy. Specifically, it first generates a negative network $\tilde{\mathcal{G}}$ via a corruption function $\tilde{\mathcal{G}} = \mathcal{C}(\mathcal{G})$. Then it uses the same encoder $\mathcal{E}$ (e.g. GCN [16]) to obtain the node embedding for the positive network $\{\mathbf{h}_1, \cdots, \mathbf{h}_N\}$ as well as the embedding for the negative network $\{\tilde{\mathbf{h}}_1, \cdots, \tilde{\mathbf{h}}_N\}$. Here, N is the number of nodes in both of the networks. The summary vector of the positive network $\mathcal{G}$ is obtained by a readout function $\mathbf{s} = \mathcal{R}(\{\mathbf{h}_1, \cdots, \mathbf{h}_N\})$ (e.g. average pooling). Finally, given $\mathbf{s}$, a discriminator $\mathcal{D}$ is used to distinguish the node embedding of the positive network $\mathbf{h}_n$ with the one from the negative network $\tilde{\mathbf{h}}_n$. For more details, please refer to [37].

Table 1: Notations

Symbols	Descriptions
$\mathcal{G}$	attributed network
$\mathcal{G}_{\mathcal{M}}$	attributed multiplex network
$\mathcal{C}$	corruption function
$\mathcal{R}$	readout function
$\mathcal{E}$	encoder function
$\mathcal{D}$	discriminator
$\mathcal{L}$	objective function
A	adjacency matrix
F	node attribute matrix
H	node embedding matrix
$I(X;Y)$	mutual information between X and Y
$H(X)$	entropy of the random variable X
$H(X\|Y)$	conditional entropy of X given Y
$\mathbf{s}$	global summary vector
$\mathbf{h}$	node embedding vector
$\mathbf{f}$	node attribute vector

Maximizing the objective function given in the following definition can effectively maximize $I(\mathbf{h}_n, \mathbf{s})$.

Definition 2.5 (Deep Graph Infomax). Given a node embedding $\mathbf{h}_n$ and a summary vector $\mathbf{s}$ of the attributed network and a negative node embedding $\tilde{\mathbf{h}}_n$, the mutual information between $\mathbf{h}_n$ and $\mathbf{s}$ can be maximized by maximizing the following objective function:

$$\mathcal{L} = \mathbb{E}[\log \mathcal{D}(\mathbf{h}_n, \mathbf{s})] + \mathbb{E}[\log(1 - \mathcal{D}(\tilde{\mathbf{h}}_n, \mathbf{s}))] \quad (3)$$

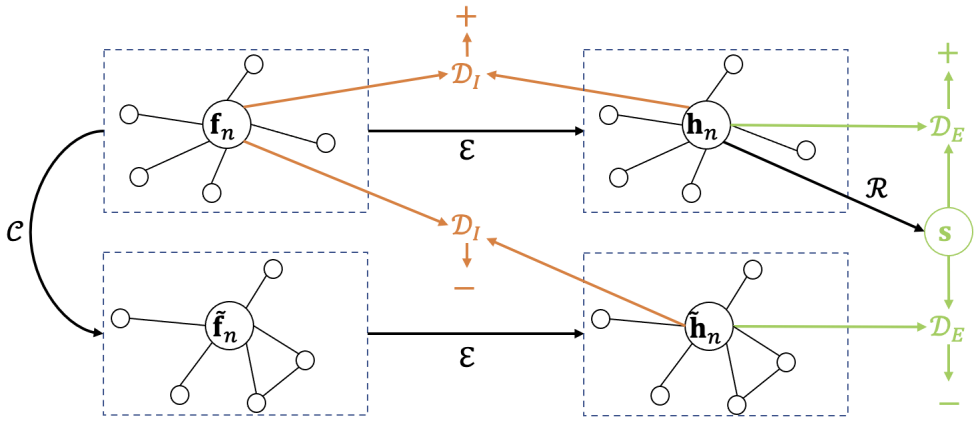
where $\mathcal{D}$ denotes the discriminator aiming at distinguish the negative node embedding $\tilde{\mathbf{h}}_n$ with the real embedding $\mathbf{h}_n$, $\mathbb{E}$ denotes the expectation.

3 METHODOLOGY

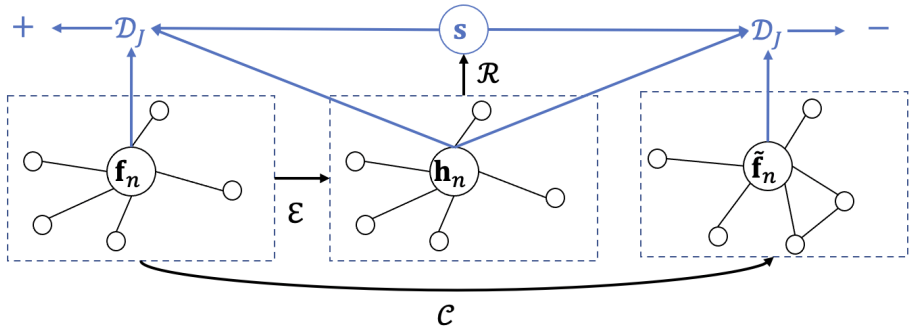
In this section, we first introduce High-order Deep Infomax (HDI) for optimizing high-order mutual information on attributed networks in Section 3.1 and then extend the proposed HDI to attributed multiplex networks by introducing High-order Deep Multiplex Infomax (HDMI) in Section 3.2.

3.1 High-order Deep Infomax on Attributed Networks

In this section, we propose a novel High-order Deep Infomax (HDI) for self-supervised learning on attributed networks based on DGI [37] and high-order mutual information (Definition 2.4). As mentioned in the introduction and preliminary (Section 2.3), DGI merely considers the *extrinsic* supervision signal: the mutual information between the node embedding $\mathbf{h}_n$ and the global summary vector $\mathbf{s}$ of a network. *Intrinsic* signal, i.e., the mutual dependence between node embedding $\mathbf{h}_n$ and attributes $\mathbf{f}_n$ has not been fully explored. We introduce high-order mutual information to simultaneously capture the extrinsic and intrinsic supervision signals as well as the synergy between them in Section 3.1.1. We describe the details of HDI in Section 3.1.2.



(a) Illustration of the extrinsic and intrinsic supervision.



(b) Illustration of the joint supervision.

Figure 2: Overview of the proposed HDI. The green part shows the extrinsic supervision which maximizes $I(\mathbf{h}_n; \mathbf{s})$. The orange part shows the intrinsic supervision which maximizes $I(\mathbf{h}_n; \mathbf{f}_n)$. The blue part illustrates the joint supervision which maximizes $I(\mathbf{h}_n; \mathbf{f}_n, \mathbf{s})$. Detailed description is presented in Section 3.1.

3.1.1 High-order Mutual Information Estimation. For the n-th node in the attributed network $\mathcal{G}$, we propose to jointly maximize the mutual information for three random variables, i.e., node embedding $\mathbf{h}_n$, the summary vector $\mathbf{s}$ and node attributes $\mathbf{f}_n$, via the high-order mutual information.

According to the definition of the high-order mutual information (Definition 2.4), when $N = 3$, we will have the following equation:

$$\begin{aligned} I(X_1; X_2; X_3) = & H(X_1) + H(X_2) + H(X_3) \\ & - H(X_1, X_2) - H(X_1, X_3) - H(X_2, X_3) \\ & + H(X_1, X_2, X_3) \end{aligned} \tag{4}$$

The above equation can be further re-written as the following equation:

$$\begin{aligned} I(X_1; X_2; X_3) = & H(X_1) + H(X_2) - H(X_1, X_2) \\ & + H(X_1) + H(X_3) - H(X_1, X_3) \\ & - H(X_1) - H(X_2, X_3) + H(X_1, X_2, X_3) \\ = & I(X_1; X_2) + I(X_1; X_3) - I(X_1; X_2, X_3) \end{aligned} \tag{5}$$

where $I(X_1; X_2, X_3)$ denotes the mutual information between the distribution of X_1 with the joint distribution of X_2 and X_3.

Therefore, by replacing the random variables X_1, X_2 and X_3 with $\mathbf{h}_n$, $\mathbf{s}$ and $\mathbf{f}_n$ we will have:

$$I(\mathbf{h}_n; \mathbf{s}; \mathbf{f}_n) = I(\mathbf{h}_n; \mathbf{s}) + I(\mathbf{h}_n; \mathbf{f}_n) - I(\mathbf{h}_n; \mathbf{s}, \mathbf{f}_n) \tag{6}$$

In the above equation, $I(\mathbf{h}_n; \mathbf{s})$ captures the extrinsic supervision signal: the mutual dependence between node embedding $\mathbf{h}_n$ and the global summary $\mathbf{s}$. $I(\mathbf{h}_n; \mathbf{f}_n)$ captures the intrinsic supervision signal: the mutual dependence between node embedding $\mathbf{h}_n$ and attributes $\mathbf{h}_n$. $I(\mathbf{h}_n; \mathbf{s}, \mathbf{f}_n)$ captures the interaction between the extrinsic and intrinsic signal.

Maximizing the high-order mutual information in Equation (6) will result in the following equation:

$$\begin{aligned} & \max I(\mathbf{h}_n; \mathbf{s}; \mathbf{f}_n) \\ = & \max(I(\mathbf{h}_n; \mathbf{s}) + I(\mathbf{h}_n; \mathbf{f}_n) - I(\mathbf{h}_n; \mathbf{s}, \mathbf{f}_n)) \\ = & \max I(\mathbf{h}_n; \mathbf{s}) + \max I(\mathbf{h}_n; \mathbf{f}_n) - \min I(\mathbf{h}_n; \mathbf{s}, \mathbf{f}_n) \\ = & \max I(\mathbf{h}_n; \mathbf{s}) + \max I(\mathbf{h}_n; \mathbf{f}_n) + \max I(\mathbf{h}_n; \mathbf{s}, \mathbf{f}_n) \end{aligned} \tag{7}$$

As can be noted in Equation (7), maximizing the high-order mutual information is equivalent to maximize the three mutual information: $I(\mathbf{h}_n; \mathbf{s})$, $I(\mathbf{h}_n; \mathbf{f}_n)$ and $I(\mathbf{h}_n; \mathbf{s}, \mathbf{f}_n)$.

Considering the fact that, during training, different mutual information on the right-hand side of Equation (7) might have different magnitudes, therefore, we use different coefficients for different mutual information. The the final objective function is given by:

$$\mathcal{L} = \lambda_E I(\mathbf{h}_n; \mathbf{s}) + \lambda_I I(\mathbf{h}_n; \mathbf{f}_n) + \lambda_J I(\mathbf{h}_n; \mathbf{s}, \mathbf{f}_n) \tag{8}$$

where λ_E, λ_I and λ_J are tunable coefficients.

Extrinsic Signal. For maximizing the extrinsic mutual dependence $I(\mathbf{h}_n; \mathbf{s})$, we follow [37] and we will have:

$$\mathcal{L}_E = \mathbb{E}[\log \mathcal{D}_E(\mathbf{h}_n, \mathbf{s})] + \mathbb{E}[\log(1 - \mathcal{D}_E(\tilde{\mathbf{h}}_n, \mathbf{s}))] \tag{9}$$

where $\mathbf{h}_n$ and $\tilde{\mathbf{h}}_n$ are the node embedding from the positive (i.e., original) $\mathcal{G}$ and the negative network $\tilde{\mathcal{G}}$ respectively; the negative network is obtained by corrupting the positive network via the corruption function C: $\tilde{\mathcal{G}} = C(\mathcal{G})$; $\mathcal{D}_E$ denotes the discriminator for distinguishing $\mathbf{h}_n$ and $\tilde{\mathbf{h}}_n$.

The green part in Figure 2a provides an illustration for the extrinsic signal.

Intrinsic Signal. For the maximization of the intrinsic mutual dependence $I(\mathbf{h}_n; \mathbf{f}_n)$, we replace $\mathbf{s}$ with $\mathbf{f}_n$ in Equation (9) and we will have:

$$\mathcal{L}_I = \mathbb{E}[\log \mathcal{D}_I(\mathbf{h}_n, \mathbf{f}_n)] + \mathbb{E}[\log(1 - \mathcal{D}_I(\tilde{\mathbf{h}}_n, \mathbf{f}_n))] \tag{10}$$

where $\mathbf{f}_n$ denotes the attribute vector of the n-th node and $\mathcal{D}_I$ denotes the discriminator.

The orange part in Figure 2a provides an illustration for the intrinsic signal.

Joint Signal. The joint signal $I(\mathbf{h}_n; \mathbf{s}, \mathbf{f}_n)$ is comprised of three random variables: $\mathbf{h}_n$, $\mathbf{s}$ and $\mathbf{f}_n$. Rather than substituting $\mathbf{s}$ with $(\mathbf{s}, \mathbf{f}_n)$ in Equation 9 and using the negative node embedding $\tilde{\mathbf{h}}_n$ to build the negative sample pairs, we propose to use the negative node attributes $\tilde{\mathbf{f}}_n$ to construct the negative samples. This is because the extrinsic signal has already captured the mutual dependence between $\mathbf{h}_n$ and $\mathbf{s}$, as well as the independence between $\tilde{\mathbf{h}}_n$ and $\mathbf{s}$ by maximizing Equation 9. The intrinsic signal has captured the mutual dependence between $\mathbf{h}_n$ and $\mathbf{f}_n$, as well as the independence between $\tilde{\mathbf{h}}_n$ and $\mathbf{f}_n$ via Equation 10. If we add a new discriminator to distinguish $\mathbf{h}_n$ from $\tilde{\mathbf{h}}_n$ given $(\mathbf{s}, \mathbf{f}_n)$, it will not bring substantial new information. Instead, we propose to let the discriminator to distinguish $(\mathbf{h}_n, \mathbf{s}, \mathbf{f}_n)$ with $(\mathbf{h}_n, \mathbf{s}, \tilde{\mathbf{f}}_n)$. By introducing negative samples of attributes $\tilde{\mathbf{f}}_n$, the encoder $\mathcal{E}$ can better capture the joint mutual dependence among $\mathbf{h}_n$, $\mathbf{s}$ and $\mathbf{f}_n$, especially the mutual dependence between $\mathbf{s}$ and $\mathbf{f}_n$, which is not captured by either extrinsic signal or intrinsic signal. Therefore, we have the following objective:

$$\mathcal{L}_J = \mathbb{E}[\log \mathcal{D}_J(\mathbf{h}_n, \mathbf{s}, \mathbf{f}_n)] + \mathbb{E}[\log(1 - \mathcal{D}_J(\mathbf{h}_n, \mathbf{s}, \tilde{\mathbf{f}}_n))] \tag{11}$$

where $\mathcal{D}_J$ denotes the discriminator for the joint signal.

The blue part in Figure 2b provides an illustration for the joint signal.

Training Objective. The final supervision signal is to maximize the following objective:

$$\mathcal{L} = \lambda_E \mathcal{L}_E + \lambda_I \mathcal{L}_I + \lambda_J \mathcal{L}_J \tag{12}$$

where λ_E, λ_I and λ_J are tunable coefficients.

3.1.2 Model Architecture. We elaborates the details of the model architecture in this section.

Network Encoder $\mathcal{E}$. We use a single layer GCN [16] as $\mathcal{E}$:

$$H = ReLU(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}FW) \tag{13}$$

where $\hat{A} = A + wI \in \mathbb{R}^{N \times N}$, $w \in \mathbb{R}$ is the weight of self-connection, $\hat{D}[i,i] = \sum_j \hat{A}[i,j]$; $F \in \mathbb{R}^{N \times d_F}$ and $H \in \mathbb{R}^{N \times d}$ denote attribute matrix and node embedding matrix, $W \in \mathbb{R}^{d_F \times d}$ denotes the transition matrix, $ReLU$ denotes the rectified linear unit activation function.

Readout Function $\mathcal{R}$. We use average pooling as $\mathcal{R}$:

$$\mathbf{s} = \mathcal{R}(H) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{h}_n \tag{14}$$

where $\mathbf{h}_n \in \mathbb{R}^d$ is the n-th row of H, $\mathbf{s} \in \mathbb{R}^d$ is the global summary vector.

Dicriminator $\mathcal{D}$. For $\mathcal{D}_E$ and $\mathcal{D}_I$, we follow DGI [37]:

$$\mathcal{D}_E(\mathbf{h}_n, \mathbf{s}) = \sigma(\mathbf{h}_n^T M_E \mathbf{s}) \tag{15}$$

$$\mathcal{D}_I(\mathbf{h}_n, \mathbf{f}_n) = \sigma(\mathbf{h}_n^T M_I \mathbf{f}_n) \tag{16}$$

where $M_E \in \mathbb{R}^{d \times d}$ and $M_I \in \mathbb{R}^{d \times d_F}$ are parameter matrices, σ is the sigmoid activation function.

As for the discriminator for the joint signal $\mathcal{D}_J$, since $\mathbf{f}_n$ and $\mathbf{s}$ are not in the same space, we first project them into the same hidden space, and then use a bi-linear function to obtain the final scores.

$$\mathbf{z}_{f_n} = \sigma(W_f \mathbf{f}_n) \tag{17}$$

$$\mathbf{z}_s = \sigma(W_s \mathbf{s}) \tag{18}$$

$$\mathbf{z} = \sigma(W_z[\mathbf{z}_{f_n}; \mathbf{z}_s]) \tag{19}$$

$$\mathcal{D}_J = \sigma(\mathbf{h}_n^T M_J \mathbf{z}) \tag{20}$$

where $W_f \in \mathbb{R}^{d \times d_F}$, $W_s \in \mathbb{R}^{d \times d}$, $W_z \in \mathbb{R}^{d \times 2d}$ and $M_J \in \mathbb{R}^{d \times d}$ are parameter matrices, σ denotes the sigmoid activation function and $[;]$ denotes concatenation operation.

Corruption Function C. We follow DGI [37] and use the random permutation of nodes as the corruption function. Specifically, we randomly shuffle the rows of attribute matrix F.

3.2 High-order Deep Multiplex Infomax

In this section, we extend HDI to the multiplex network and propose a High-order Deep Multiplex Infomax (HDMI) model. An attributed multiplex network is comprised of multiple layers of attributed networks. In order to learn node embedding for the multiplex networks, we first learn node embedding on each of its layers separately and then combine them via a fusion module (Section 3.2.1). We leverage the high-order mutual information to train the fusion module (Section 3.2.2).

3.2.1 Fusion of Embedding. The simplest way of combining node embedding from different layers is average pooling. However, different layers are related to each other. Therefore, we use the semantic attention [43] based method to fuse the node embedding, as shown in Figure 3.

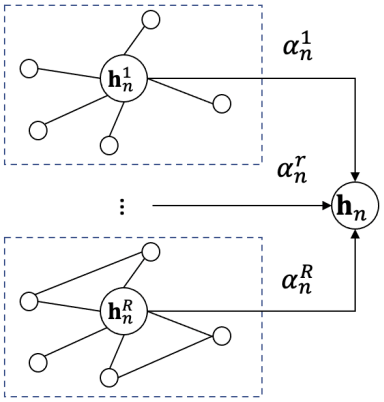


Figure 3: Illustration of the fusion module.

Given a node embedding from the r-th layer $\mathbf{h}_n^r \in \mathbb{R}^d$, we use layer-dependent semantic attention to obtain its score α_n^r by:

$$\alpha_n^r = \tanh(\mathbf{y}^{r\,T} V_r \mathbf{h}_n^r) \tag{21}$$

where $V_r \in \mathbb{R}^{d' \times d}$ is the parameter matrix, $\mathbf{y}^r$ denotes the hidden representation vector of the r-th layer, tanh denotes the tangent activation function.

Then the weights of node embedding from different layers are given by:

$$\alpha_n^r = \frac{\exp(\alpha_n^r)}{\sum_{r'=1}^{R} \exp(\alpha_n^{r'})} \tag{22}$$

where R is the number of layers in the multiplex network.

The final embedding of the n-th node can be obtained by:

$$\mathbf{h}_n = \sum_{r=1}^{R} \alpha_n^r \mathbf{h}_n^r \tag{23}$$

3.2.2 Training. To train the fusion module, we propose to use high-order mutual information introduced in Section 3.1. Given the fused embedding $\mathbf{h}_n$, we maximize the mutual information between $\mathbf{h}_n$ with its attributes $\mathbf{f}_n$ and global summary vector $\mathbf{s}$ of the multiplex network. The training objective and model architecture are the same as those introduced in Section 3.1, but there are two differences: the global summary vector $\mathbf{s}$ and negative node embedding $\tilde{\mathbf{h}}_n$. For $\mathbf{s}$ of the multiplex network, we use average pooling over the fused embedding $\mathbf{h}_n$ to obtain it. Additionally, we obtain the negative node embedding $\tilde{\mathbf{h}}_n$ of the multiplex network by combining the negative node embedding $\tilde{\mathbf{h}}_n^r$ from different layers via the fusion module.

The fusion module can be trained jointly with HDI on different layers, and the final objective of HDMI is:

$$\mathcal{L} = \lambda_M \mathcal{L}_M + \sum_{r=1}^{R} \lambda_r \mathcal{L}_r \tag{24}$$

where $\mathcal{L}_M$ and $\mathcal{L}_r$ denote the objective for the fusion module and the r-th layer respectively, λ_M and λ_r are tunable coefficients. Note that both $\mathcal{L}_M$ and $\mathcal{L}_r$ are given by Equation (12).

4 EXPERIMENTS

We presents the experiments to answer the following questions:

- Q1 How will the HDI and HDMI improve the quality of the learned node embedding?
- Q2 Will the fusion module assign proper attention scores to different layers?

4.1 Experimental Setup

4.1.1 Datasets. We use the same datasets as [28].

ACM. The ACM[1] dataset contains 3,025 papers with two types of paper relations: paper-author-paper and paper-subject-paper. The attribute of each paper is a 1,830-dimensional bag-of-words representation of the abstract. The nodes are categorized into three classes: Database, Wireless Communication and Data Mining. When training the classifier, 600 nodes are used as training samples.

IMDB. The IMDB[2] dataset contains 3,550 movies with two types of relations, including movie-actor-movie and movie-director-movie. The attribute of each movie is a 1,007-dimensional bag-of-words representation of its plots. The nodes are annotated with Action, Comedy or Drama, and 300 nodes are used for training classifiers.

DBLP. The DBLP[3] dataset [35] is a multiplex network of 7,907 papers. There are three types of relations: paper-paper, paper-author-paper, paper-author-term-author-paper. The attribute of each paper is a 2,000-dimensional bag-of-words representation of its abstracts. The nodes can be categorized into four categories: Data Mining, Artificial Intelligence, Computer Vision and Natural Language Processing. 80 nodes are used for training classifiers.

Amazon. The Amazon[4] dataset [11] contains 7,621 items from four categories (Beauty, Automotive, Patio Lawn and Garden, and Baby) with three types of relations (Also View, Also Bought and Bought Together). The attribute of each item is a 2,000-dimensional bag-of-words representation of its description.

4.1.2 Comparison Methods. We compare our proposed method with two sets of baseline methods: network embedding methods and multiplex network embedding methods.

Network Embedding.

- Methods disregarding attributes: DeepWalk [30] and node2vec [9]. These two methods are random-walk and skip-gram based embedding models.
- Methods considering attributes: GCN [16] and GAT [36]. These two methods learn node embedding based on the local structure of the nodes, and the best performance of the two methods are reported. DGI [37] maximizes the mutual information between node embedding and the global summary vector. ANRL [46] uses skip-gram to model the local contextual topology and uses an auto-encoder to capture attribute information. CAN [24] learns node embedding and attribute embedding in the same semantic space. DGCN [49] considers both local and global consistency.

[1] https://www.acm.org/
[2] https://www.imdb.com/
[3] https://aminer.org/AMinerNetwork
[4] https://www.amazon.com/

Table 2: Statistics of the datasets

Datasets	# Nodes	Relation Types	# Edges	# Attributes	# Labeled Data	# Classes
ACM	3,025	Paper-Subject-Paper (PSP) Paper-Author-Paper (PAP)	2,210,761 29,281	1,830 (Paper Abstract)	600	3
IMDB	3,550	Movie-Actor-Movie (MAM) Movie-Director-Movie (MDM)	66,428 13,788	1,007 (Movie plot)	300	3
DBLP	7,907	Paper-Author-Paper (PAP) Paper-Paper-Paper (PPP) Paper-Author-Term-Author-Paper (PATAP)	144,783 90,145 57,137,515	2,000 (Paper Abstract)	80	4
Amazon	7,621	Item-AlsoView-Item (IVI) Item-AlsoBought-Item (IBI) Item-BoughtTogether-Item (IOI)	266,237 1,104,257 16,305	2,000 (Item description)	80	4

Multiplex Network Embedding.

- Methods disregarding attributes: CMNA [5] uses the cross-network information to refine both of the inter-network and intra-network node embedding. MNE [44] uses a common embedding with multiple additional embedding from different layers for each node.
- Methods considering attributes: mGCN [21] and HAN [39] use GCN/GAT to extract node embedding for each layer of the multiplex networks and then combine them via attention mechanism. DMGI and $\text{DMGI}_{\text{attn}}$ [28] extend DGI onto multiplex networks and uses consensus regularization to combine node embedding from different layers, where $\text{DMGI}_{\text{attn}}$ leverages attention to obtain the reference node embedding.

4.1.3 Evaluation Metrics. We evaluate our proposed HDMI and comparison methods on both of the unsupervised tasks (i.e., clustering and similarity search [28, 39]) and a supervised task (i.e., node classification) with the following evaluation metrics.

- Macro-F1 and Micro-F1 are used to evaluate models on the node classification task.
- Normalized Mutual Information (NMI) is adopted for the node clustering task.
- Sim@5 evaluates how well does a model project nodes with the same label to nearby positions in the embedding space.

For the clustering and classification, we first train models with their self-supervision signals and then run the clustering methods (i.e., K-means) and classification methods (i.e., logistic regression) to obtain the NMI and Macro-F1/Micro-F1 scores. For similarity search, we follow [28, 39] to first compute the cosine similarity between each pair of nodes based on their embedding. Then for each node, the top-5 most similar nodes are selected to calculate the ratio of nodes sharing the same label (Sim@5).

4.1.4 Implementation Details. We set the dimension of node embedding as 128, and use Adam optimizer [14] with the learning rate of 0.001 to optimize the models. The weight of the self-connection for GCN is fixed as 3. Following [28], for HDMI, we use the same discriminator for different layers. We use grid search to tune the coefficients and report the best results. Early stopping with a patience of 100 is adopted to prevent overfitting.

4.2 Quantitative Evaluation

In this section, we present the experimental results on both supervised and unsupervised downstream tasks to quantitatively demonstrate the effectiveness of the proposed supervision signals as well as the proposed models HDI and HDMI.

4.2.1 Overall Performance. We present the experimental results of node classification, node clustering and similarity search on the multiplex networks in Table 3 and Table 4. HDI separately learns embedding on different layers and uses average pooling to combine node embedding from different layers. Table 3 shows that HDMI outperforms the state-of-the-art models for all of the supervised tasks, and Table 4 shows that HDMI achieves higher scores for most of the cases on the unsupervised tasks. These results demonstrate that the embedding extracted by HDMI is more discriminative. Additionally, the scores of HDI are generally competitive to the state-of-the-art methods, which demonstrate the effectiveness of the proposed HDI to a certain degree.

4.2.2 Ablation Study.

Performance of the fusion module. To evaluate the proposed fusion module, we compare it with average pooling. As shown in Table 3 and Table 4, HDMI outperforms HDI on all metrics for all datasets, except for the Sim@5 on the ACM dataset. However, it can be noted that the gap is very tiny.

Performance of different supervision signals. We compare the performance of the Extrinsic (E.), Intrinsic (I.), and Joint (J) signal as well as the Reconstruction (R.) error on attributed networks (different layers of the multiplex networks) in the datasets, and present experimental results in Table 5 and Table 6. Firstly, incorporating the mutual dependence between embedding and attributes could improve the model's performance on both supervised and unsupervised tasks, which can be observed by comparing E. with E.+R. and E.+I. Secondly, maximizing mutual information between node embedding and attributes (E.+I.) is better than minimizing the reconstruction error of attributes given node embedding (E.+R.). Thirdly, the joint signal (E.+I.+J.) can further improve the discriminative ability of node embedding for the node classification task. Finally, combining node embedding from different layers (lower parts of Table 5-6) will result in better results, indicating that different layers can mutually help each other.

Table 3: Overall performance on the supervised task: node classification.

Dataset	ACM		IMDB		DBLP		Amazon	
Metric	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
DeepWalk	0.739	0.748	0.532	0.550	0.533	0.537	0.663	0.671
node2vec	0.741	0.749	0.533	0.550	0.543	0.547	0.662	0.669
GCN/GAT	0.869	0.870	0.603	0.611	0.734	0.717	0.646	0.649
DGI	0.881	0.881	0.598	0.606	0.723	0.720	0.403	0.418
ANRL	0.819	0.820	0.573	0.576	0.770	0.699	0.692	0.690
CAN	0.590	0.636	0.577	0.588	0.702	0.694	0.498	0.499
DGCN	0.888	0.888	0.582	0.592	0.707	0.698	0.478	0.509
CMNA	0.782	0.788	0.549	0.566	0.566	0.561	0.657	0.665
MNE	0.792	0.797	0.552	0.574	0.566	0.562	0.556	0.567
mGCN	0.858	0.860	0.623	0.630	0.725	0.713	0.660	0.661
HAN	0.878	0.879	0.599	0.607	0.716	0.708	0.501	0.509
DMGI	0.898	0.898	0.648	0.648	0.771	0.766	0.746	0.748
$DMGI_{attn}$	0.887	0.887	0.602	0.606	0.778	0.770	0.758	0.758
HDI	**0.901**	0.900	0.634	0.638	0.814	0.800	0.804	0.806
HDMI	**0.901**	**0.901**	**0.650**	**0.658**	**0.820**	**0.811**	**0.808**	**0.812**

Table 4: Overall performance on the unsupervised tasks: node clustering and similarity search.

Dataset	ACM		IMDB		DBLP		Amazon	
Metric	NMI	Sim@5	NMI	Sim@5	NMI	Sim@5	NMI	Sim@5
DeepWalk	0.310	0.710	0.117	0.490	0.348	0.629	0.083	0.726
node2vec	0.309	0.710	0.123	0.487	0.382	0.629	0.074	0.738
GCN/GAT	0.671	0.867	0.176	0.565	0.465	0.724	0.287	0.624
DGI	0.640	0.889	0.182	0.578	0.551	0.786	0.007	0.558
ANRL	0.515	0.814	0.163	0.527	0.332	0.720	0.166	0.763
CAN	0.504	0.836	0.074	0.544	0.323	0.792	0.001	0.537
DGCN	0.691	0.690	0.143	0.179	0.462	0.491	0.143	0.194
CMNA	0.498	0.363	0.152	0.069	0.420	0.511	0.070	0.435
MNE	0.545	0.791	0.013	0.482	0.136	0.711	0.001	0.395
mGCN	0.668	0.873	0.183	0.550	0.468	0.726	0.301	0.630
HAN	0.658	0.872	0.164	0.561	0.472	0.779	0.029	0.495
DMGI	0.687	0.898	0.196	0.605	0.409	0.766	0.425	0.816
$DMGI_{attn}$	**0.702**	**0.901**	0.185	0.586	0.554	0.798	0.412	0.825
HDI	0.650	0.900	0.194	0.605	0.570	0.799	0.487	0.856
HDMI	0.695	0.898	**0.198**	**0.607**	**0.582**	**0.809**	**0.500**	**0.857**

4.3 Qualitative Evaluation

In this section, we present the qualitative evaluation results.

4.3.1 *Visualization of node embedding.*

Node embedding of an attributed network. We show the t-sne [22] visualization of the node embedding of the IOI layer in the Amazon dataset learned by the Extrinsic (E.), the Extrinsic + Reconstruction (E. + R.), the Extrinsic + Intrinsic (E. + I.) and the Extrinsic + Intrinsic + Joint (E. + I. + J.) signals in Figure 4. Different colors in the figure represent different classes. It is obvious that the intrinsic signal and the joint signal significantly improve the discriminative ability of the node embedding, and the joint signal can further improve the quality of the node embedding Additionally, Figure 4b shows that the reconstruction error does not substantially connect with the discriminative ability of the node embedding.

Node embedding of a multiplex network. The t-sne visualization for the node embedding learned by HDI on the PSP and the PAP layer of the ACM multiplex network are presented in Figure 5a-5b. Figure 5c-5d show the combined embedding obtained by average pooling and fusion module. As can be noted in the red boxes of Figure 5b-5c, average pooling for the embedding of different layers better separates nodes than the embedding learned from PAP layer alone. Additionally, Figure 5c-5d show that the fusion module separates the nodes even better than the average pooling.

4.3.2 *Attention scores.*

We present the visualization of attention scores and the Micro-F1 scores for each layer of the multiplex networks in Figure 6. Generally, for the layers that have higher Micro-F1 scores, their attention scores are also higher, demonstrating the effectiveness of the proposed fusion module.

Table 5: Ablation study on the supervised task: node classification. MaF1 and MiF1 denote Macro-F1 and Micro-F1. E., I., J., and R. denote the Extrinsic, Intrinsic, Joint mutual information and Reconstruction error. The results of $HDMI_{avg}$ and HDMI are copied from Table 3 to better show the effectiveness of combining different layers.

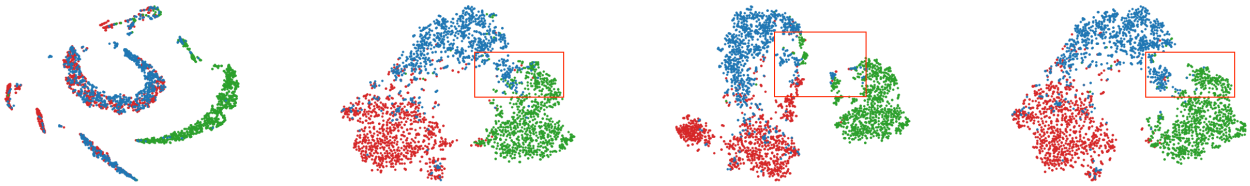
Dataset	ACM				IMDB				DBLP						Amazon					
Layer	PSP		PAP		MDM		MAM		PAP		PPP		PATAP		IVI		IBI		IOI	
Metric	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1
E.	0.663	0.668	0.855	0.853	0.573	0.586	0.558	0.564	0.804	0.796	0.728	0.717	0.240	0.272	0.380	0.388	0.386	0.410	0.569	0.574
E. + R.	0.668	0.673	0.864	0.847	0.590	0.597	0.560	0.570	0.809	0.801	0.737	0.728	0.240	0.280	0.392	0.398	0.410	0.427	0.579	0.589
E. + I.	0.719	0.732	0.886	0.887	0.617	0.624	0.593	0.600	0.803	0.792	0.742	0.733	0.240	0.276	0.559	0.561	0.517	0.527	0.792	**0.799**
E. + I. + J.	**0.742**	**0.744**	**0.889**	**0.888**	**0.626**	**0.631**	**0.600**	**0.606**	**0.812**	**0.803**	**0.751**	**0.745**	**0.241**	**0.284**	**0.581**	**0.583**	**0.524**	**0.529**	**0.796**	**0.799**
Metric	MaF1		MiF1		MaF1		MiF1		MaF1		FiF1				MaF1				MiF1	
HDI	**0.901**		0.900		0.634		0.638		0.814		0.800				0.804				0.806	
HDMI	**0.901**		**0.901**		**0.650**		**0.658**		**0.820**		**0.811**				**0.808**				**0.812**	

Table 6: Ablation study on the unsupervised tasks: node clustering and similarity search. S@5 denotes Sim@5. E., I., J., and R. denote the Extrinsic, Intrinsic, Joint mutual information and Reconstruction error. The results of $HDMI_{avg}$ and HDMI are copied from Table 4 to better show the effectiveness of combining different layers.

Dataset	ACM				IMDB				DBLP						Amazon					
Layer	PSP		PAP		MDM		MAM		PAP		PPP		PATAP		IVI		IBI		IOI	
Metric	NMI	S@5	NMI	S@5	NMI	S@5	NMI	S@5	NMI	S@5	NMI	S@5	NMI	S@5	NMI	S@5	NMI	S@5	NMI	S@5
E.	0.526	0.698	0.651	0.872	0.145	0.549	0.089	0.495	0.547	0.800	0.404	0.741	0.054	0.583	0.002	0.395	0.003	0.414	0.038	0.701
E. + R.	0.525	**0.728**	0.659	0.874	0.150	0.552	0.079	0.490	0.564	0.804	**0.421**	0.741	0.051	0.568	0.002	0.399	0.003	0.426	0.020	0.660
E. + I.	0.527	0.708	0.656	0.882	0.193	**0.595**	**0.143**	**0.527**	**0.569**	0.802	0.405	0.741	0.053	0.569	0.152	0.512	0.143	0.517	0.401	0.824
E. + I. + J.	**0.528**	0.716	**0.662**	**0.886**	**0.194**	0.592	**0.143**	**0.527**	0.562	**0.805**	0.408	**0.742**	**0.054**	**0.591**	**0.169**	**0.544**	**0.153**	**0.525**	**0.407**	**0.826**
Metric	NMI		Sim@5		NMI		Sim@5		NMI		Sim@5				NMI				Sim@5	
HDI	0.650		**0.900**		0.194		0.605		0.570		0.799				0.487				0.856	
HDMI	**0.695**		0.898		**0.198**		**0.607**		**0.582**		**0.809**				**0.500**				**0.857**	

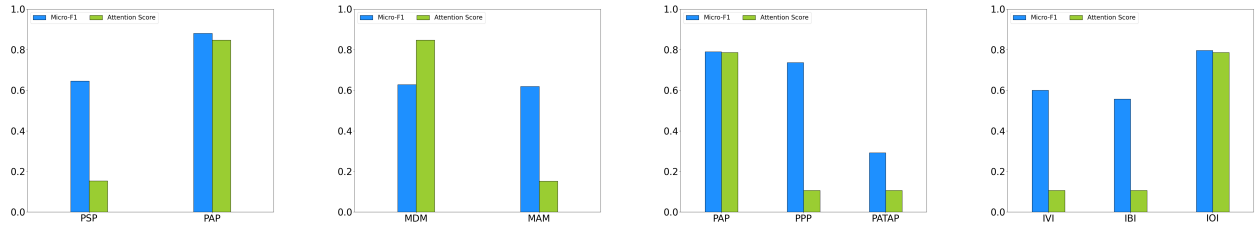
(a) E. **(b) E. + R.** **(c) E. + I.** **(d) E. + I. + J.**

Figure 4: Visualization of the node embedding on the IOI layer of the Amazon dataset. The four different colors represent four different classes of the nodes. E., I., J., and R. denote the Extrinsic, Intrinsic, Joint mutual information and Reconstruction error.



(a) PSP **(b) PAP** **(c) Average** **(d) Fusion**

Figure 5: Visualization of the node embedding on the ACM dataset. PSP and PAP are two layers of the ACM multiplex network. Different colors represent different classes of the nodes. Average and fusion denote using average pooling and the proposed fusion module to obtain the combined embedding. As shown in the red boxes, the fusion module better separates different classes than average pooling.



(a) ACM (b) IMDB (c) DBLP (d) Amazon

Figure 6: Attention scores and Micro-F1 scores for each layer.

5 RELATED WORKS

In this section, we briefly review the most relevant works, including the self-supervised graph representation learning (Section 5.1), the graph neural networks (Section 5.2), as well as the mutual information methods (Section 5.3).

5.1 Self-supervised Network Representation Learning

In this section, we mainly focus on self-supervised network embedding methods. For a comprehensive review, please refer to [6, 45, 48]. We leave the mutual information related methods to Section 5.2, and graph neural networks with other signals to Section 5.3.

Inspired by word2vec [25], deepwalk [30] and node2vec [9] leverage random walks to sample context of a node, and then learn node embedding by maximizing the probabilities of the sampled neighbors. SDNE [38] and LINE [34] propose to learn node embedding based on the first-order and the second-order proximities between nodes. Struct2vec [38] further designs a training signal based on the local structural similarity of the nodes. However, these methods ignore node attributes. Zhang et al. [46] uses a neighbor enhancement auto-encoder to encoder attributes with the re-construction error. Meng et al. [24] propose to co-embed attributes and nodes into the same semantic space via Variational Auto-Encoder (VAE) [15], but VAE tends to *minimize* the mutual information between inputs and hidden features [47].

The multiplex network is also known as multi-dimensional network [20]. Liu et al. [19] propose a random walk based method to map multiplex networks into a vector space. Zhang et al. [44] propose a scalable multiplex network embedding method trained via negative sampling. Shi et al. [33] leverage preservation and collaboration to learn node embedding. Ma et al. [21] propose mGCN for multiplex networks trained via negative sampling. Cen et al. [4] use the random walk based method to train a unified network for attributed multiplex networks. Ban et al. [42] introduce a density metric as the training signal to improve the performance of node clustering. Li et al. [17] introduce MANE considering both within-layer and cross-layer dependency.

5.2 Graph Neural Networks

In this section, we briefly review the graph neural networks with external training signals (e.g. cross-entropy between predicted labels and real labels). For a comprehensive review of the graph neural networks, please refer to [45, 48].

Kipf et al. [16] propose Graph Convolutional Network (GCN) based on [7]. Hamilton et al. [10] propose an aggregation-based model called GraphSAGE. Graph Attention Network (GAT) [36] learns different weights for a node's neighbors by attention mechanism. Zhuang et al. [49] propose DGCN which considers both local and global consistency.

Qu et al. [31] use an attention mechanism to embed multiplex networks into a single collaborated embedding. Wang et al. [39] propose an attention based method called HAN for merging node embedding from different layers. Chu et al. [5] propose CrossMNA which leverages cross-network information for network alignment. Yan et al. [40] introduce DINGAL for dynamic knowledge graph alignment. Jing et al.[13] propose TGCN for modeling high-order tensor graphs.

5.3 Mutual Information Methods

Belghazi et al. [2] propose MINE to estimate mutual information of two random variables by neural networks. Mukherjee et al. [26] propose a classifier based neural estimator for conditional mutual information. Recently, the infomax principle [18] has been used for self-supervised representation learning in computer vision [12] and speech recognition [32] by maximizing the mutual information between different hidden features. In the field of network representation learning, DGI [37] maximizes the mutual information of node embedding with the global summary vector. Peng et al. [29] propose GMI for homogeneous networks. Park et al. [27, 28] extend DGI onto multiplex networks and propose a consensus regularization to combine embedding of different layers.

6 CONCLUSION

In this paper, we introduce a novel High-order Deep Multiplex Infomax (HDMI) to learn network embedding for multiplex networks via self-supervised learning. Firstly, we propose a High-order Deep Infomax (HDI) and use high-order mutual information as the training signal for attributed networks. The proposed signal simultaneously captures the extrinsic signal (i.e., the mutual dependence between node embedding and the global summary), the intrinsic signal (i.e., the mutual dependence between node embedding and attributes), and the interaction between these two signals. Secondly, we propose a fusion module based on the attention mechanism to combine node embedding from different layers. We evaluate the proposed HDMI on four real-world datasets for both unsupervised and supervised downstream tasks. The results demonstrate the effectiveness of the proposed HDMI, HDI, and the fusion module.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under grant No. 1947135, by the NSF Program on Fairness in AI in collaboration with Amazon under award No. 1939725, and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1C1C1009081) The content of the information in this document does not necessarily reflect the position or the policy of the Government or Amazon, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. 2018. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062* (2018).

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).

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

[5] Xiaokai Chu, Xinxin Fan, Di Yao, Zhihua Zhu, Jianhui Huang, and Jingping Bi. 2019. Cross-network embedding for multi-network alignment. In *The World Wide Web Conference*. 273–284.

[6] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering* 31, 5 (2018), 833–852.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*. 3844–3852.

[8] Hongchang Gao and Heng Huang. 2018. Deep Attributed Network Embedding.. In *IJCAI*, Vol. 18. New York, NY, 3364–3370.

[9] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD*. 855–864.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[11] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[12] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).

[13] Baoyu Jing, Hanghang Tong, and Yada Zhu. 2021. Network of Tensor Time Series. In *The World Wide Web Conference*. https://doi.org/10.1145/3442381.3449969

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[16] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

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

[18] Ralph Linsker. 1988. Self-organization in a perceptual network. *Computer* 21, 3 (1988), 105–117.

[19] Weiyi Liu, Pin-Yu Chen, Sailung Yeung, Toyotaro Suzumura, and Lingli Chen. 2017. Principled multilayer network embedding. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 134–141.

[20] Yao Ma, Zhaochun Ren, Ziheng Jiang, Jiliang Tang, and Dawei Yin. 2018. Multi-dimensional network embedding with hierarchical structure. In *Proceedings of the eleventh ACM WSDM*. 387–395.

[21] Yao Ma, Suhang Wang, Chara C Aggarwal, Dawei Yin, and Jiliang Tang. 2019. Multi-dimensional graph convolutional networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 657–665.

[22] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[23] William McGill. 1954. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory* 4, 4 (1954), 93–111.

[24] Zaiqiao Meng, Shangsong Liang, Hongyan Bao, and Xiangliang Zhang. 2019. Co-embedding attributed networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 393–401.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[26] Sudipto Mukherjee, Himanshu Asnani, and Sreeram Kannan. 2020. CCMI: Classifier based conditional mutual information estimation. In *Uncertainty in Artificial Intelligence*. PMLR, 1083–1093.

[27] Chanyoung Park, Jiawei Han, and Hwanjo Yu. 2020. Deep multiplex graph infomax: Attentive multiplex network embedding using global information. *Knowledge-Based Systems* 197 (2020), 105861.

[28] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. 2020. Unsupervised Attributed Multiplex Network Embedding. In *AAAI*. 5371–5378.

[29] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph Representation Learning via Graphical Mutual Information Maximization. In *Proceedings of The Web Conference 2020*. 259–270.

[30] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD*. 701–710.

[31] Meng Qu, Jian Tang, Jingbo Shang, Xiang Ren, Ming Zhang, and Jiawei Han. 2017. An attention-based collaboration framework for multi-view network representation learning. In *Proceedings of the 2017 ACM CIKM*. 1767–1776.

[32] Mirco Ravanelli and Yoshua Bengio. 2018. Learning speaker representations with mutual information. *arXiv preprint arXiv:1812.00271* (2018).

[33] Yu Shi, Fangqiu Han, Xinwei He, Xinran He, Carl Yang, Jie Luo, and Jiawei Han. 2018. mvn2vec: Preservation and collaboration in multi-view network embedding. *arXiv preprint arXiv:1801.06597* (2018).

[34] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[35] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD*. 990–998.

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[37] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep graph infomax. *arXiv preprint arXiv:1809.10341* (2018).

[38] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 1225–1234.

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

[40] Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. 2021. Dynamic Knowledge Alignment. In *AAAI*.

[41] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. 2019. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4066–4075.

[42] Ban Yikun, Liu Xin, Huang Ling, Duan Yitao, Liu Xue, and Xu Wei. 2019. No place to hide: Catching fraudulent entities in tensors. In *The World Wide Web Conference*. 83–93.

[43] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4651–4659.

[44] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. 2018. Scalable Multiplex Network Embedding.. In *IJCAI*, Vol. 18. 3082–3088.

[45] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. 2019. Graph convolutional networks: a comprehensive review. *Computational Social Networks* 6, 1 (2019), 11.

[46] Zhen Zhang, Hongxia Yang, Jiajun Bu, Sheng Zhou, Pinggang Yu, Jianwei Zhang, Martin Ester, and Can Wang. 2018. ANRL: Attributed Network Representation Learning via Deep Neural Networks.. In *IJCAI*, Vol. 18. 3155–3161.

[47] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262* (2017).

[48] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434* (2018).

[49] Chenyi Zhuang and Qiang Ma. 2018. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 World Wide Web Conference*. 499–508.