Deep Contrastive Multiview Network Embedding

Mengqi Zhang*, Yanqiao Zhu*, Qiang Liu, Shu Wu†, and Liang Wang
[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences
{mengqi.zhang, yanqiao.zhu}@cripac.ia.ac.cn, {qiang.liu, shu.wu, wangliang}@nlpr.ia.ac.cn

ABSTRACT

Multiview network embedding aims at projecting nodes in the network to low-dimensional vectors, while preserving their multiple relations and attribute information. Contrastive learning approaches have shown promising performance in this task. However, they neglect the semantic consistency between fused and view representations and have difficulty in modeling complementary information between different views. To deal with these deficiencies, this work presents a novel Contrastive leaRning framEwork for Multiview network Embedding (CREME). In our work, different views can be obtained based on the various relations among nodes. Then, we generate view embeddings via proper view encoders and utilize an attentive multiview aggregator to fuse these representations. Particularly, we design two collaborative contrastive objectives, view fusion InfoMax and inter-view InfoMin, to train the model in a self-supervised manner. The former objective distills information from embeddings generated from different views, while the latter captures complementary information among views to promote distinctive view embeddings. We also show that the two objectives can be unified into one objective for model training. Extensive experiments on three real-world datasets demonstrate that our proposed CREME is able to consistently outperform state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies → Unsupervised learning**.

ACM Reference Format:
Mengqi Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, and Liang Wang. 2022. Deep Contrastive Multiview Network Embedding. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557577

1 INTRODUCTION

Real-world networks often consist of various types of relations, which are known as multiview networks or multiplex networks. Take academic networks as an example: Two nodes denoting papers are connected if they have common authors or they cite each other. Multiview network embedding aims at projecting nodes in the network to low-dimensional vectors, while preserving their multiple relations and attribute information [2, 11, 14, 28]. Since it is usually labor-extensive to manually collect high-quality labels, obtaining informative embeddings without supervision for multiview networks has attracted a lot of attention in the community.

So far, a series of self-supervised methods have been proposed for multiview network embedding. Some early approaches [2, 14, 28] mainly focus on the compression of multiple graph views but ignore node attributes. To capture the attribute and structure information together, some others [1, 10] combine graph neural networks and relational reconstruction tasks for self-supervised learning. However, most of these methods over-emphasize the network proximity, thus limiting the expressiveness of learned embeddings [13, 15, 23]. Inspired by visual representation learning [6], recent studies attempt to introduce contrastive learning into multiview networks [7, 11] and have achieved compelling performance.

However, we argue that these contrastive models still have two deficiencies. Firstly, their contrastive strategies neglect the semantic consistency between views in the original network. In the paradigm of multiview network embedding, the final node embedding is usually obtained by aggregating node embeddings from different views induced by relations. Based on the hypothesis that a powerful representation is one that models view-invariant factors [18, 19], the fused embedding should capture sufficient semantic information shared among multiple relations. In contrast, the existing models focus on contrasting node- and graph-level embeddings, while ignoring capturing view-invariant factors in relation-induced views. As a result, the fused representation suffers from limited expressiveness. Secondly, these contrastive methods fail to further consider inter-view dependency, leading to suboptimal performance. Consider that in multiview networks, node representations obtained from different views tend to be similar due to the shared node attributes. To improve the discriminative ability of node embeddings, it is thus vital to capture the complementary information of different views [17].

To deal with the two aforementioned challenges, we present a novel deep Contrastive leaRning framEwork for Multiview network Embedding, CREME for brevity. The overall framework of CREME is presented in Figure 1. Specifically, we first generate views according to various relations of multiview networks. Then, we obtain each view representation via a view encoder based on graph attention networks (§2.4). Next, we combine all the relations and form a fusion view. Accordingly, we introduce a multiview aggregator to integrate different view representations as the final node representations (§2.5). To enable self-supervised training, we propose a novel contrasting strategy (view fusion InfoMax) with a regularization term (inter-view InfoMin) (§2.3). The first objective *maximizes the mutual information between the fused representation*

*These authors made equal contribution to this work.
†To whom correspondence should be addressed.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557577

and view representations to promote multiview fusion, while the second objective enforces *information minimization among graph views*, which improves distinctiveness of view representations, so as to preserve complementary information among relation-induced views. We further show that the two contrastive objectives can be collectively combined into one elegant, unified objective function.

The main contributions of this work are summarized as follows: Firstly, we propose a novel contrastive framework CREME for multiview network embedding, the core of which contains two collaborative contrastive objectives, view fusion InfoMax and inter-view InfoMin. Secondly, we conduct extensive empirical studies on three real-world datasets. The results demonstrate the effectiveness of CREME over state-of-the-art baselines.

2 THE PROPOSED METHOD

2.1 Preliminaries

Definition (Multiview networks). A multiview network is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X, \phi)$ whose edges are associated with more than one types. In such a network, the mapping $\phi : \mathcal{E} \to \mathcal{R}, |\mathcal{R}| > 1$ associates an edge with an edge type; $\mathcal{V}, \mathcal{E} \in \mathcal{V} \times \mathcal{V}, X \in \mathbb{R}^{|\mathcal{V}| \times F}$, and $\mathcal{R}$ represents the node set, the edge set, the node attribute matrix, and the set of edge types respectively.

In this work, we consider the task of self-supervised multiview network embedding, where we aim to learn a d-dimensional ($d \ll F$) vector z_i representing for each node i without accessing to labels.

2.2 The Overall Framework

The overall framework of CREME is illustrated in Figure 1. There are three main components: (1) a view encoder that projects nodes in each relation-induced view into low-dimensional representations, (2) a multiview aggregator, which adaptively integrates view representations and obtains the final fused node embeddings for $\mathcal{G}$, and (3) a unified contrastive objective to enable self-supervised learning of the view encoder and the multiview aggregator.

Our CREME framework follows the common multiview contrastive learning paradigm, which essentially seeks to maximize the agreement of representations among different views. Different from traditional graph contrastive learning methods [5, 24, 27, 31, 32], our graph views are naturally induced by different relations rather than generated by data augmentations.

After obtaining views according to relations, we utilize an encoding function $f_r : A_r \times X \to Z^r \in \mathbb{R}^{N \times d}$ for view $\mathcal{G}_r$ to obtain relation view representations. Thereafter, we employ a multiview aggregator $g : (Z^1, ..., Z^{|\mathcal{R}|}) \to Z \in \mathbb{R}^{N \times d}$ to obtain the fused representations for multiview network $\mathcal{G}$. Here, z_i^r in Z^r is the representation of node i in view $\mathcal{G}_r$ and z_i in Z is the representation of node i in graph $\mathcal{G}$, which can be regarded as a fusion view of the original network. In this following subsections, we introduce the learning objective at first (§2.3) and then proceed to the design of view encoders (§2.4) and multiview aggregators (§2.5) in detail.

2.3 Contrastive Objectives

2.3.1 View fusion InfoMax. At first, we propose a novel contrasting strategy to train the model by maximizing the semantic consistency

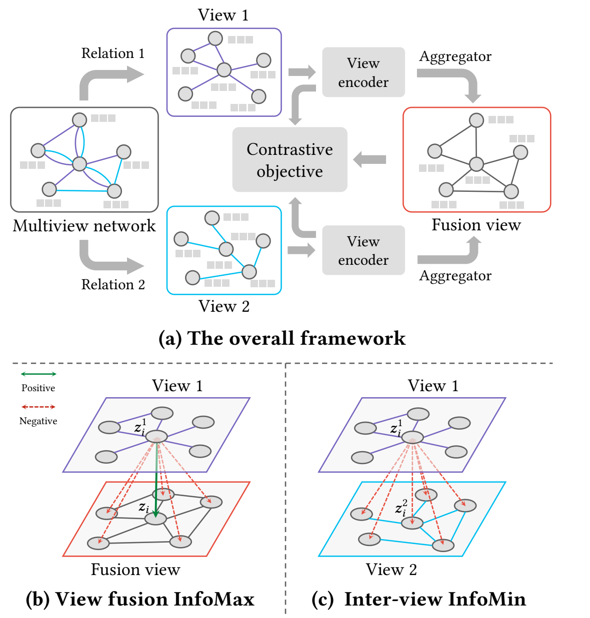


Figure 1: An overview of CREME: Figure (a) presents the framework of our proposed model; Figures (b) and (c) illustrate the contrastive strategies between different graph views, where the green solid arrow indicates positive pairs and the orange dashed arrows indicate negative pairs.

of view representation Z^r in each view $\mathcal{G}_r$ and the fusion representation Z of $\mathcal{G}$. Following mutual information estimation [20], this can be achieved by maximizing the Mutual Information (MI) between Z^r and Z. In this way, the resulting fusion view representation can *selectively* distill information of each relation view.

Specifically, for an anchor node i, its view representation and the fused representation (z_i^r, z_i) constitutes a positive pair. Following prior studies [30–32], we set all other nodes in two graph views as negative pairs of z_i^r. We illustrate the view fusion InfoMax in Figure 1(b). Formally, the objective of view fusion InfoMax is defined as

$$\mathcal{L}_o(z_i^r, z_i) = \log \frac{e^{\theta(z_i^r, z_i)/\tau}}{e^{\theta(z_i^r, z_i)/\tau} + \sum_{j \neq i} e^{\theta(z_i^r, z_j)/\tau} + \sum_{j \neq i} e^{\theta(z_i^r, z_j^r)/\tau}}, \tag{1}$$

where $\theta(u, v) = s(p(u), p(v))$ is a critic function, $s(\cdot, \cdot)$ is implemented using a simple cosine similarity, $p(\cdot)$ is a non-linear projection function to enhance the expression power of the critic function, and τ is a temperature parameter. For simplicity, we denote the denominator in Eq. (1) as $\rho(z_i^r, z_i)$ hereafter:

$$\rho(z_i^r, z_i) = e^{\theta(z_i^r, z_i)/\tau} + \sum_{j \neq i} e^{\theta(z_i^r, z_j)/\tau} + \sum_{j \neq i} e^{\theta(z_i^r, z_j^r)/\tau}. \tag{2}$$

2.3.2 Inter-view InfoMin. The previous objective only focuses on the relationship between each relation view and the final fusion view. Considering that in our multiview network embedding setting, each node shares the same node attribute in different relation views, and thus their view embeddings tend to be similar during view encoding. Therefore, we propose the second objective to further regularize information among relation views. Our approach is to add a regularization term to minimize the MI of relation view representations, so as to enforce the model to learn discriminative view representations.

Instead of directly optimizing MI between Z^r and Z^k for any view pair $(r,k)_{r\neq k}$, we simply set $(z_i^r, z_i^k)_{r\neq k}$ and $(z_i^r, z_j^k)_{i\neq j}$ as additional negative samples in Eq. (1), as illustrated in Figure 1(c). In this way, we elegantly combine the two contrastive objectives:

$$\mathcal{L}(z_i^r, z_i) = \log \frac{e^{\theta(z_i^r, z_i)/\tau}}{\rho(z_i^r, z_i) + \sum_{j\in\mathcal{G}_k} \mathbb{1}_{[k\neq r]} e^{\theta(z_i^r, z_j^k)/\tau}}. \quad (3)$$

2.3.3 Learning objectives. Finally, the overall objective is defined as an average of MI over all positive pairs, formally given by

$$\mathcal{J} = \frac{1}{N\cdot|\mathcal{R}|} \sum_{i=1}^{N} \sum_{r=1}^{|\mathcal{R}|} \mathcal{L}(z_i^r, z_i). \quad (4)$$

To summarize, the view fusion InfoMax objective enforces the multiview aggregator to adaptively distill information from each relation view. The inter-view InfoMin regularization further constrains different relation view representations to be distinct to each other, which makes the model capture the complementary information of contained in each relation.

2.4 View Encoders

For the input multiview network, we generate views, each according to one provided relation. In this way, we essentially convert the heterogeneous network into a series of homogeneous networks. Then, for each relation-induced view, we capture the structural and attribute information of nodes through a view-specific graph attention neural network. To be specific, we leverage the self-attention mechanism [22] to compute the weight coefficients α_{ij}^r between node i and its neighbor j in $\mathcal{G}_r$:

$$\alpha_{ij}^r = \frac{\exp(\sigma(\boldsymbol{a}_r^\top [\boldsymbol{M}_r \boldsymbol{x}_i \parallel \boldsymbol{M}_r \boldsymbol{x}_j]))}{\sum_{k\in\mathcal{N}_i^r} \exp(\sigma(\boldsymbol{a}_r^\top [\boldsymbol{M}_r \boldsymbol{x}_i \parallel \boldsymbol{M}_r \boldsymbol{x}_k]))}, \quad (5)$$

where $\mathcal{N}_i^r$ is the set of neighbors of node i in view $\mathcal{G}_r$, $\boldsymbol{a}_r \in \mathbb{R}^{2d}$ is a view-specific weight vector, $\boldsymbol{M}_r \in \mathbb{R}^{d\times F}$ is a transformation matrix projecting each node attribute into the corresponding semantic space, and $\sigma(\cdot) = \mathrm{ReLU}(\cdot)$ is the non-linear function. The view representation of node i in $\mathcal{G}_r$ can then be calculated by aggregating their neighbor node embeddings:

$$z_i^r = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i^r} \alpha_{ij}^{r,k} \boldsymbol{M}_r \boldsymbol{x}_j\right). \quad (6)$$

Here, we leverage a multihead attention mechanism of K heads [22], where each attention head has its own independent encoder parameters and $\alpha_{ij}^{r,k}$ is the attention coefficient in the k-th head.

2.5 Multiview Aggregators

After obtaining each relation view representations, we leverage a multiview aggregator to integrate semantic information from all relation-induced views for each node in $\mathcal{G}$. To preserve important information during aggregation, we utilize an another attention network to aggregate the embeddings of different views for each node. The importance of each of view embedding z_i^r can be calculated by

$$w_i^r = \boldsymbol{q}^\top \tanh(\boldsymbol{W} z_i^r + \boldsymbol{b}), \quad (7)$$

Table 1: Statistics of datasets used in experiments.

Dataset	Relations	#Nodes	#Edges	#Attributes	#Classes
ACM	P–S–P P–A–P	3,025	2,210,761 29,281	1,830	3
IMDB	M–A–M M–D–M	3550	66,428 13,788	1,007	3
DBLP	P–A–P P–P–P P–A–T–A–P	7,907	144,783 90,145 57,137,515	2,000	4

where $\boldsymbol{q} \in \mathbb{R}^d$ denotes the attention vector, $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ is the weight matrix parameter, and $\boldsymbol{b} \in \mathbb{R}^d$ is the bias vector. For node i, the weight of each view embedding z_i^r can be obtained by:

$$\beta_i^r = \frac{\exp(w_i^r)}{\sum_{r=1}^{|\mathcal{R}|} \exp(w_i^r)}. \quad (8)$$

Then, the fused representation is obtained by taking weighted average of view representations:

$$z_i = \sum_{r=1}^{|\mathcal{R}|} \beta_i^r z_i^r. \quad (9)$$

These fused representations can be used for downstream tasks.

3 EXPERIMENTS

In this section, we conduct experiments on three real-world datasets to evaluate our proposed method.

3.1 Experimental Setup

◇ **Datasets.** We conduct experiments on ACM[1], IMDB[2], and DBLP[3]. ACM contains item nodes with two types of relations: Paper–Author–Paper (P–A–P) and Paper–Subject–Paper (P–S–P). IMDB contains movie nodes with Movie–Actor–Movie (M–A–M) and Movie-Director-Movie (M–D–M) relations. DBLP has paper nodes with Paper–Paper–Paper (P–P–P), Paper–Author–Paper (P–A–P), and Paper–Author–Term–Author–Paper (P–A–T–A–P) relations. For fair comparison, we follow the same data preprocessing as in DMGI [11] for all datasets, whose statistics are shown in Table 1.

◇ **Baselines.** The baselines include (a) homogeneous network models are DeepWalk [12], Node2Vec [4], ANRL [29], GCN [9], GAT [22], DGI [24], and GraphCL [27], (b) heterogeneous network models Metapath2vec [3] and HAN [25], and (c) multiview network models MNE [28], GATNE [1], DMGI [11], and HDMI [7].

◇ **Implementation details.** For all methods, we set the embedding size to 64 and default to the recommended hyperparameters settings. For CREME, we use the Adam optimizer [8] with the initial learning rate to 0.001, the weight decay to 1e-5, the temperature τ to 0.7, and set the dropout of view encoder to 0.6. We implement our CREME in MindSpore. Following DMGI [11], we run the evaluation for 50 times and report the averaged performance. We use a logistic regression and a k-Means model to perform node classification and node clustering on the learned embeddings, respectively. We use Macro-F1 (MaF1) and Micro-F1 (MiF1) as metrics for node

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

Table 2: Performance comparison of different models. The highest and second-to-best performance is highlighted in boldface and underlined respectively.

Method	ACM			IMDB			DBLP		
	MaF1	MiF1	NMI	MaF1	MiF1	NMI	MaF1	MiF1	NMI
Deepwalk	0.739	0.748	0.310	0.532	0.55	0.117	0.533	0.537	0.348
node2vec	0.741	0.749	0.309	0.533	0.55	0.123	0.543	0.547	0.382
ANRL	0.819	0.820	0.515	0.573	0.576	0.163	0.770	0.699	0.332
GCN/GAT	0.869	0.870	0.671	0.603	0.611	0.176	0.734	0.717	0.465
DGI	0.881	0.881	0.640	0.598	0.606	0.182	0.723	0.720	0.551
GraphCL	0.892	0.894	0.656	0.613	0.624	0.183	0.736	0.722	0.562
Metapath2vec	0.752	0.758	0.314	0.546	0.574	0.144	0.653	0.649	0.382
HAN	0.878	0.879	0.658	0.599	0.607	0.164	0.716	0.708	0.472
MNE	0.792	0.797	0.545	0.552	0574	0.013	0.566	0.562	0.136
GATNE	0.846	0.841	0.521	0.494	0.504	0.048	0.673	0.665	0.436
DMGI	0.898	0.898	0.687	0.648	0.648	0.196	0.771	0.766	0.409
DMGI-attn	0.887	0.887	0.702	0.602	0.606	0.185	0.778	0.770	0.554
HDMI	0.895	0.894	0.657	0.601	0.610	0.197	0.805	0.795	0.544
CREME	**0.907**	**0.906**	**0.726**	**0.672**	**0.675**	**0.211**	**0.812**	**0.798**	**0.623**

classification, and Normalized Mutual Information (NMI) for node clustering.

3.2 Performance Comparison

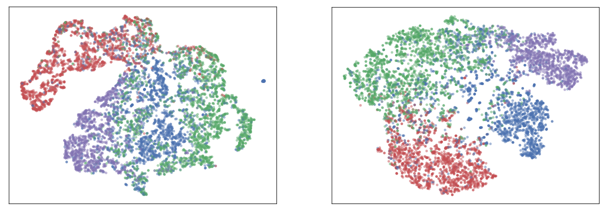
We first report the performance of all compared methods on node classification and node clustering tasks. Table 2 summaries the results. Our CREME consistently achieves the best performance on three datasets. Compared with the strong baselines DMGI and HDMI, CREME obtains the most noticeable performance improvement. This verifies that our framework has strong capabilities to utilize different graph views. CREME is also competitive with semi-supervised models, i.e., HAN, GAT, and GCN, which shows the superiority of our framework in the training of view encoder and multiview aggregator. Traditional baselines MNE and Metapath2vec are inferior to that of attribute-aware network methods, such as HAN, ANRL, DMGI, and HDMI, on most datasets. This indicates that the node attributes are necessary for multiview network embedding. Furthermore, most multiview methods, such as HDMI, DMGI, GATNE, and MNE, generally outperform single-view methods. This verifies the necessity of modeling multiple relations. GraphCL and DGI, as the contrastive learning methods, perform the best among single-view network embedding methods in most datasets. This result demonstrates the superiority of mutual information optimization over proximity reconstruction in representation learning.

3.3 Ablation Studies

To investigate the effects of the contrastive objectives, view encoder, and multiview aggregator, we compare CREME with five variants. *CRE_V-mean* and *CRE_V-max* set the operator as mean and max in the view encoder, respectively. *CRE_M-mean* and *CRE_M-max* set the operator as mean and max in the multiview aggregator, respectively. *CRE_C-ori* excludes the inter-view InfoMin objective. The results are shown in Table 3. From the table, we see that CRE_V-mean and CRE_V-max perform worse than CRE_M-mean and CRE_M-max on most datasets, especially for node clustering, which suggests that the view encoder plays a more important role compared to the multiview aggregator. The performance of CRE_M-mean and CRE_M-max is not significantly different from that of CREME in ACM and IMDB. However, the performance of CRE_M-max is worse in DBLP. The reason is that DBLP data is more complicated than ACM and IMDB, as shown by the fact that DBLP have more relations. The max aggregator tends to ignore multiplicities than the attention and mean aggregator [26]. CREME outperforms CRE_C-ori in most cases, which demonstrates that our inter-view InfoMin could supplement the view fusion InfoMax objective.

Table 3: Performance of different model variants.

Variant	ACM			IMDB			DBLP		
	MaF1	MiF1	NMI	MaF1	MiF1	NMI	MaF1	MiF1	NMI
CRE_V-mean	0.786	0.778	0.394	0.519	0.546	0.056	0.801	0.795	0.516
CRE_V-max	0.824	0.828	0.529	0.551	0.562	0.015	0.810	0.796	0.516
CRE_M-mean	0.896	0.896	0.714	0.672	0.673	0.196	0.803	0.783	0.623
CRE_M-max	0.905	0.899	0.723	0.671	0.674	0.203	0.792	0.780	0.606
CRE_C-ori	0.894	0.893	0.725	0.657	0.661	**0.216**	0.795	0.775	0.519
CREME	**0.907**	**0.906**	**0.726**	**0.672**	**0.675**	0.211	**0.812**	**0.798**	**0.623**



(a) HDMI (b) CREME

Figure 2: Visualization of the learned node embedding by HDMI and CREME on DBLP.

3.4 Visualization

To provide a qualitative evaluation, we map the node embedding of the DBLP network learned by CREME and HDMI into a 2D space using the t-SNE algorithm [21] and plot them in Figure 2. We find that CREME exhibit more distinct boundaries and clusters than HDMI. Moreover, the Silhouette scores [16] of the embeddings obtained by HDMI and CREME are 0.11 and 0.28 (the higher, the better), respectively, which once again verifies that CREME can learn informative node embeddings.

4 CONCLUSION

In this work, we have proposed a novel contrastive learning framework for unsupervised learning of multiview networks. In our framework, we propose two contrastive objectives through optimizing mutual information between different views, fusion view InfoMax and inter-view InfoMin, which distills information from every relation view and promotes discriminative view representations. Extensive experiments on three real-world multiview networks verify the effectiveness of CREME.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (62141608 and U19B2038) and CAAI–Huawei MindSpore Open Fund.

REFERENCES

[1] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation learning for attributed multiplex heterogeneous network. In *KDD*.

[2] Xiaokai Chu, Xinxin Fan, Di Yao, Zhihua Zhu, Jianhui Huang, and Jingping Bi. 2019. Cross-network embedding for multi-network alignment. In *WWW*.

[3] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*.

[4] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.

[5] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *ICML*.

[6] 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. In *ICLR*.

[7] Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. Hdmi: High-order deep multiplex infomax. In *Proceedings of the Web Conference 2021*.

[8] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. (2015).

[9] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[10] Yao Ma, Zhaochun Ren, Ziheng Jiang, Jiliang Tang, and Dawei Yin. 2018. Multi-dimensional network embedding with hierarchical structure. In *WSDM*.

[11] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. 2020. Unsupervised attributed multiplex network embedding. In *AAAI*.

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.

[13] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*.

[14] 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 *CIKM*.

[15] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *KDD*.

[16] Peter J. Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* (1987).

[17] Yu Shi, Fangqiu Han, Xinran He, Carl Yang, Jie Luo, and Jiawei Han. 2018. mvn2vec: Preservation and Collaboration in Multi-View Network Embedding. *arXiv.org* (2018). arXiv:1801.06597v3 [cs.SI]

[18] Linda Smith and Michael Gasser. 2005. The development of embodied cognition: Six lessons from babies. *Artif. Life* (2005).

[19] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *ECCV*.

[20] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv.org* (2018). arXiv:1807.03748v2 [cs.LG]

[21] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* (2008).

[22] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[23] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep Graph Infomax. In *ICLR*.

[24] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR*.

[25] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *WWW*.

[26] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *ICLR*.

[27] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *NeurIPS*.

[28] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. 2018. Scalable multiplex network embedding. In *IJCAI*.

[29] 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*.

[30] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An Empirical Study of Graph Contrastive Learning. In *NeurIPS Datasets and Benchmarks*.

[31] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. In *GRL+@ICML*.

[32] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*.