An Effective and Robust Framework by Modeling Correlations of Multiplex Network Embedding

1st Pengfei Jiao
Law school
Tianjin University
TianJin, China
pjiao@tju.edu.cn

2nd Ruili Lu
Tianjin International Engineering Institute
Tianjin University
TianJin, China
luruili@tju.edu.cn

3rd Di Jin
College of Intelligence and Computing
Tianjin University
TianJin, China
jindi@tju.edu.cn

4th Yinghui Wang
College of Intelligence and Computing
Tianjin University
TianJin, China
wangyinghui@tju.edu.cn

5th Huaming Wu
Center for Applied Mathematics
Tianjin University
TianJin, China
whming@tju.edu.cn

***Abstract*—The dependencies across different layers are an important property in multiplex networks and a few methods have been proposed to learn the dependencies in various ways. When capturing the dependencies across different layers, some of them assumed the structure among layers following consistent connectivity to force two nodes with a link in one layer tend to have links in other layers, some introduced a common vector to model the shared information across all layers. However, the correlations among layers in multiplex networks are diverse, which go beyond the connectivity consistency. In this paper, we propose a novel Modeling Correlations for Multiplex network Embedding (MCME) framework to learn the robust node representations for each layer. It can deal with complex correlations with a common structure, layer similarity and node heterogeneity through a unified framework in multiplex networks. To evaluate our proposed model, we conduct extensive experiments on several real-world datasets and the results demonstrate that our proposed model consistently outperforms state-of-the-art methods.**

***Index Terms*—Multiplex Network, Multiplex Network Embedding, Link Prediction, Layer correlations, Node correlations**

I. Introduction

In recent years, with the rapid expansion of network data volume and access, data representation in the form of multiplex networks is becoming an increasingly common practice. Generally speaking, a multiplex network is composed of a set of nodes and different types of connections. Each type of link and its nodes constitute a layer with special functions and structure for the multiplex network. With their powerful representation and modeling capabilities, multiplex networks have a variety of research tasks and applications. Compared with a single or homogeneous network, certain network tasks based on this network, e.g., community detection and link prediction, pose new challenges. One of the most important is the complex dependencies across the different layers.

To model the dependency in multiplex network, there have been a few studies on multiplex network embedding, e.g., matrix factorization-based [1], [2], random walk-based [3], [4] and deep neural network-based [5]–[7]. For the ones modeling the correlations among layers, [8], [9] think nodes in all layers tend to have a consistent connectivity structure and force node embeddings for the same node in different layers closer to share the consistent structure. [4], [6] model the cross-layer node pairs which also lead to two nodes linked in one layer are more likely to link with each other in another layer. [3], [5] introduce a common vector to model the shared information across all layers in multiplex networks. [7] require some prior knowledge to select the most informative layer or the hierarchy dependencies between the layers while this prior knowledge is sometimes difficult to obtain.

However, in many real-life multiplex networks, the semantics between the layers are usually different, or even opposite, and the statistical characteristics of the same node across layers are significantly different. Then the correlations among layers are not only the connectivity consistency or the shared common information of the same nodes in different layers, and nodes in different layers have specific and diverse structure so that links in one layer may not exist in another layer, e.g., in a social network, the structure between the networks constructed by the types of following and messaging may be more similar than the structure between the messaging and comment, or for the networks constructed by the types of following and blacklist relationships, in which the users to follow and to blacklist are not completely different. The existing approaches mentioned above have lost the ability to model the complex correlations and are not well applicable to these networks with large structural differences.

For clarity, we take a popular multiplex network from CKM data [10] as an example. Fig. 1(a) shows the 7th node and its ego network of all three layers and we take a case to predict the top-5 links for node 7 in each layer. We can see that the edges of node 7 existing in the second layer are also likely to exist in the first layer, while the third layer does not actually have the same edge as the other two layers. The predicting results of different models on this dataset are shown in Fig. 1(c). It can

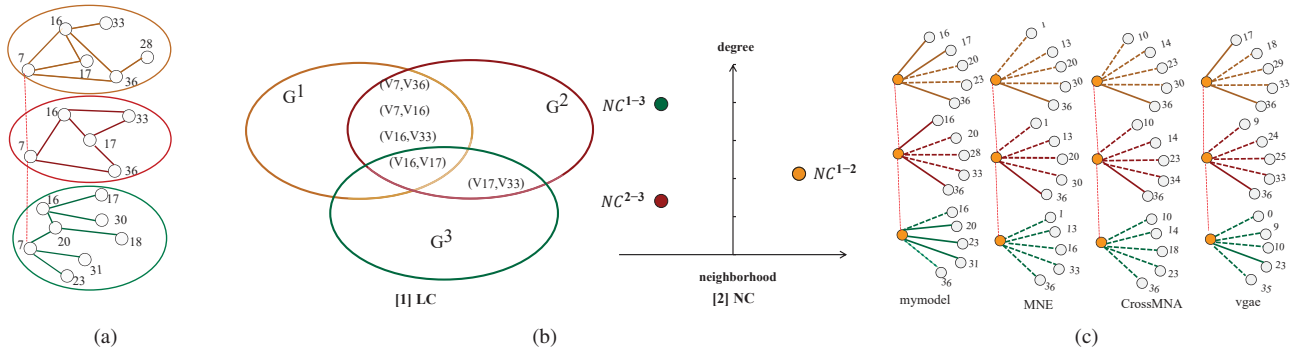

Fig. 1. (a) shows node 7 and its ego network of all three layers based on CKM dataset. (b) is a simple diagram of Layer Correlation and Node correlation based on (a). (c) intuitively shows the results of our model and several baseline models on predicting top-5 links for node 7 in each layer.

be seen that in the third layer, MNE [3] and CrossMNA [5] tend to predict the edges that exist in the other two layers even if the third layer does not actually have the same edge as the other two layers. As a single-layered embedding method, vgae [11] only focuses on the structural information within the layer. Even if both two links of node 7 in the second layer exist in the first layer, the prediction of each layer is independent of other layers. In contrast to the poor results of these algorithms, our proposed model can model the complex dependencies across the layers and is completely accurate for predicting the real edges in each layer.

The great superiority of this model is mainly because the link correlations in multiplex networks are not just the connectivity consistency or shared structure, but also more complex. Modeling complex link correlations can make our model more robust and not just applicable to the networks following the connectivity consistency. We depict them from two perspectives. From the layered perspective, each layer in a multiplex network has its own unique semantics. The semantics of some layers are relatively close to each other, while the semantics of other layers may be very different or even opposite. Then, we define a Layer Correlation (LC) index, which is measured by the intersection of the edge sets of different layers. As shown in Fig. 1(b), the correlation between G^1 and G^2 is more related than G^3 and G^1 (or G^2). More subtly, from the node perspective, the correlations of nodes between layers are also varying. In some layers, the structure of nodes is similar, while in other layers, it may be considerably different. Correspondingly, we define a Node Correlation (NC) index to weigh the difference of degrees and neighborhoods of nodes in different layers. As shown in Fig. 1(b), the correlation of node 7 between the first layer and the second layer is positive, while the correlation between the third layer and the second layer is negative. With the constraints of Layer Correlation and Node Correlation, we enforce the vectors with greater correlations to be closer and with weaker or negative correlations to be further.

In summary, our major contributions can be listed as follows:

- We investigate the existing multiplex network embedding methods and find that when modeling the dependencies across different layers, some models need necessary prior knowledge about the relationship between layers, while some models consider nodes to be consistent in connectivity or think there is some shared information between anchor nodes. However, from the datasets in real life, we find that the correlations are more complex and multiple aspects, e.g., varying degrees and different directions (positive and negative).
- We divide the correlations in multiplex networks into layer correlation and node correlation. And we define two indexes to separately formalize their degrees of correlation.
- We propose a novel **M**odeling **C**orrelations for **M**ultiplex network **E**mbedding (MCME) framework, which incorporates the complex link correlations in multiplex networks into the node representation learning process of each layer. Extensive experiments on several multiplex networks prove the effectiveness of our model.

II. Proposed Method

A. Framework and Overview

In a multiplex network, the different semantic meanings at each layer will lead to diverse structures. As shown in Fig. 1(a), the connections of node v_7 vary in each layer. However, different types of connectivity relations share the same set of nodes. Therefore, as the same entity, they display some common features across the networks. For example, as shown in Fig. 1(a), the connections of node v_7 that exist in the second layer are very likely to exist in the first layer. However, the correlation in a multiplex network is not just the consistency of connectivity between nodes. For instance, the connections of node v_7 in the third layer do not actually have the same connection as the other two layers. The correlations are complex and multiple aspects and modeling the complex link correlations can make the model more generalized and not just applicable to the networks that follow the connectivity consistency. Therefore, we propose a model MCME to model the link correlations in multiplex networks. Fig. 2 shows the overall framework of MCME.

MCME is mainly composed of three components, namely, learning the common features among nodes in all layers

by encoding on the aggregated network regardless of the relation types, generating the node representations in each layer by combining common vectors and layer vectors, and integrating complex correlations, including constraints of layer correlations and node correlations. The overall objective of our approach is to minimize the following loss function:

$$\mathcal{L} = loss_{agg} + loss_{el} + \alpha \cdot loss_{lc} + \beta \cdot loss_{nc}. \tag{1}$$

Next, we will introduce the details of each part.

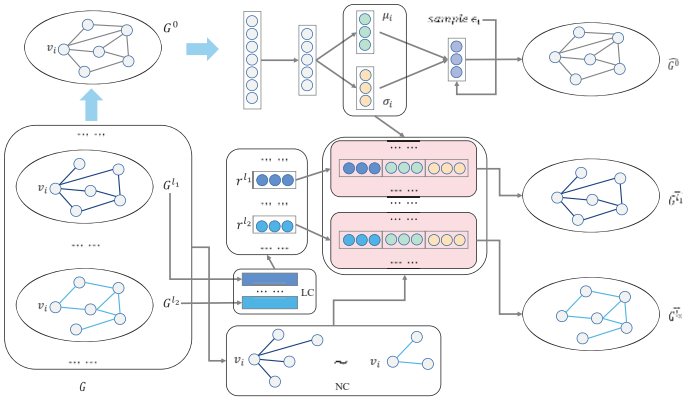

Fig. 2. The proposed MCME framework.

B. Encoder on Aggregated Networks

As the same entity in different layers of a multiplex network, nodes share some common features across the networks. To leverage the common information among nodes in the multiplex network, we define common vectors $z \in R^{d_1}$ for nodes to preserve the common features and it is learned in a new aggregated network. Firstly, we aggregate a single-layered network $G^0 = (V, \xi^0)$, by merging the edges in all layers regardless of the type. Then, we get the node representations in the aggregated network G^0 by an inference model:

$$q(\boldsymbol{Z} \mid \boldsymbol{I}, \boldsymbol{A^0}) = \prod_{i=1}^{N} q(z_i \mid \boldsymbol{I}, \boldsymbol{A^0}), \tag{2}$$

$$q(z_i \mid \boldsymbol{I}, \boldsymbol{A^0}) = \mathcal{N}(z_i \mid \mu_i, diag(\sigma_i^2)), \tag{3}$$

where μ_i and σ_i are learning by a two-layer GCN [12]:

$$\boldsymbol{\mu} = GCN_{\boldsymbol{\mu}}(\boldsymbol{I}, \boldsymbol{A^0}), \tag{4}$$

$$\boldsymbol{\sigma} = GCN_{\boldsymbol{\sigma}}(\boldsymbol{I}, \boldsymbol{A^0}), \tag{5}$$

$$GCN(\boldsymbol{I}, \boldsymbol{A^0}) = \tilde{\boldsymbol{A}}^{\boldsymbol{0}} ReLU(\tilde{\boldsymbol{A}}^{\boldsymbol{0}} \boldsymbol{I} \boldsymbol{W}_0) \boldsymbol{W}_1, \tag{6}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the matrices of mean vectors μ_i and variance vectors σ_i, respectively. $\boldsymbol{W}_0$ and $\boldsymbol{W}_1$ are weight matrices, GCN_μ and GCN_σ share the first-layer parameters $\boldsymbol{W}_0$. $\tilde{\boldsymbol{A}}^{\boldsymbol{0}} = \boldsymbol{D}^{\boldsymbol{0}^{-\frac{1}{2}}}(\boldsymbol{A^0} + \boldsymbol{I})\boldsymbol{D}^{\boldsymbol{0}^{-\frac{1}{2}}}$ is the symmetrically normalized adjacency matrix, $\boldsymbol{D^0}$ is the degree matrix of G^0 and $\boldsymbol{I}$ is the identity matrix.

With the object of reconstructing the adjacency matrix, the node distributions with $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ should be similar to the standard Gaussian, the loss function is defined as:

$$\begin{aligned} loss_{agg} = -\Big[& \sum_{(v_i^0, v_j^0) \in \xi^0} \log(p(v_i^0, v_j^0) + \\ & \sum_{(v_i^0, v_j^0) \in \xi_{neg}^0} (1 - \log(p(v_i^0, v_j^0))\Big] \\ & - KL\left[q(\boldsymbol{Z} \mid \boldsymbol{I}, \boldsymbol{A^0}) \parallel p(\boldsymbol{Z})\right], \end{aligned} \tag{7}$$

where $p(v_i^0, v_j^0)$ denotes the edge probability between v_i and v_j in G^0, and ξ_{neg}^0 is all the node pairs that are not in ξ^0. $KL\left[q(\cdot) \parallel p(\cdot)\right]$ is the Kullback-Leibler divergence between $q(\cdot)$ and $p(\cdot)$, $p(\boldsymbol{Z}) = \prod_i p(\boldsymbol{z}_i) = \prod_i \mathcal{N}(\boldsymbol{z}_i \mid 0, \boldsymbol{I})$ is a Gaussian prior.

We use the inner product to denote the edge probability between v_i and v_j, which is defined as follows:

$$p(v_i^0, v_j^0) = \sigma(\boldsymbol{z}_i^T \boldsymbol{z}_j), \tag{8}$$

where $\sigma(\cdot)$ is the logistic sigmoid function.

C. Generate Node Representations in Each Layer

The different semantics at each layer in the multiplex network will lead to diverse structures. To capture the unique characteristics of each layer in G, we define layer vectors to extract the semantics of the network, i.e., $r^l \in R^{d_2}$ for the network of the $l-th$ layer. Combining the node distribution with $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ learned in the above section with layer vector, we can get the node representations in each layer, i.e., node v_i the $l-th$ layer, as follows:

$$z_i^l = \mathcal{N}(\boldsymbol{z}_i \mid \mu_i, diag(\sigma_i^2)) \oplus r^l, \tag{9}$$

where $z_i^l \in R^d$ and $\oplus$ is the sum operation. Their dimensions must be equal, i.e., $d_1 = d_2 = d$, and then we have z_i^l as follows:

$$z_i^l = \mathcal{N}(\boldsymbol{z}_i \mid \mu_i, diag(\sigma_i^2)) + r^l. \tag{10}$$

Through the combination, Z^l represents the node embedding matrix in each layer, which contains not only the commonness among nodes, but also the unique characteristics of the selected network.

With the node representations in each layer, we reconstruct the adjacency matrix of each layer by minimizing the following loss function:

$$\begin{aligned} loss_{el} = -\sum_{l=1}^{L} \Big[& \sum_{(v_i^l, v_j^l) \in \xi^l} \log(p(v_i^l, v_j^l) + \\ & \sum_{(v_i^l, v_j^l) \in \xi_{neg}^l} (1 - \log(p(v_i^l, v_j^l))\Big], \end{aligned} \tag{11}$$

where ξ_{neg}^l is all the node pairs that are not in ξ^l. $p(v_i^l, v_j^l)$ denotes the edge probability between v_i and v_j in layer l calculated by the internal product, the same as the previous section:

$$p(v_i^l, v_j^l) = \sigma({\boldsymbol{z}_i^l}^T \boldsymbol{z}_j^l). \tag{12}$$

It should be noted that due to the sparsity of network in real life, the number of negative edges is very large and can generally be linearly bounded by the number of nodes. It is computationally expensive to consider all the pairs in loss functions defined in Eqs.7 and 11. To solve this problem, we use the negative sampling approach [13]. For each $(v_i, v_j) \in \xi^l$, we randomly sample k nodes that are not connected to node v_i in layer l. These k samples are put into the set of negative samples. In this way, the size of negative samples is only k times as large as that of positive samples.

D. Modeling Correlations

We divide the correlations in the multiplex network into layer correlation and node correlation and define two indicators to measure the correlation degree, respectively.

1) Modeling Layer Correlations: We use the global overlap rate between two layers for quantification to formalize layer correlations.

$$\mathcal{LC}^{l_1 l_2} = \frac{\mid \xi^{l_1} \cap \xi^{l_2} \mid}{\mid \xi^{l_1} \cup \xi^{l_2} \mid}, \tag{13}$$

where l_1 and l_2 are the layers, ξ^{l_1} and ξ^{l_2} are their respective edge sets, and $\mid \cdot \mid$ is the size of the set. The range of $\mathcal{LC}$ is [0, 1]. The larger the value of $\mathcal{LC}$, the more similar the network structure of the two layers. If the network structures of the two layers are completely different, then $\mathcal{LC} = 0$.

We enforce the layer vectors closer to the larger layer correlations among layers by the following terms:

$$loss_{lc} = \sum_{l_1=1}^{L} \sum_{l_2=l_1+1}^{L} \mid\mid r^{l_1} - r^{l_2} \mid\mid \mathcal{LC}^{l_1 l_2}. \tag{14}$$

2) Modeling Node Correlations: For node correlations, we mainly consider two important properties of nodes: degree and neighborhoods. For the correlation in the degree of node v_i between layers l_1 and l_2, we have:

$$\mathcal{NC}_d^{l_1 l_2} = e^{-|d(v_i^{l_1}) - d(v_i^{l_2})|}, \tag{15}$$

where $\mid \cdot \mid$ is an absolute value, $d(v_i^{l_1})$ and $d(v_i^{l_2})$ denote the degree of node v_i in layers l_1 and l_2, respectively. The greater the difference between the degrees of the nodes in the two layers, the smaller the correlation.

For the correlation in neighborhoods of node v_i between layers l_1 and l_2, we have:

$$\mathcal{NC}_n^{l_1 l_2} = \frac{1}{1 + e^{-|\mathcal{N}(v_i^{l_1}) \cap \mathcal{N}(v_i^{l_2})|}}, \tag{16}$$

where $\mathcal{N}(v_i^{l_1})$ and $\mathcal{N}(v_i^{l_2})$ denote the neighborhoods of node v_i in layers l_1 and l_2, respectively. The larger the size of the intersection of node neighborhoods in the two layers, the larger the correlation. Finally, the node correlation index of node v_i can be expressed as:

$$\mathcal{NC}_i^{l_1 l_2} = \begin{cases} -\frac{1}{1+e^{-|d(v_i^{l_1}) - d(v_i^{l_2})|}}, & if \mid \mathcal{N}(v_i^{l_1}) \cap \mathcal{N}(v_i^{l_2}) \mid = 0, \\ \frac{2\mathcal{NC}_d^{l_1 l_2} \mathcal{NC}_n^{l_1 l_2}}{\mathcal{NC}_d^{l_1 l_2} + \mathcal{NC}_n^{l_1 l_2}}, & otherwise. \end{cases} \tag{17}$$

If the size of the intersection of node neighborhoods in the two layers equals zero, it indicates that the node structures of these two layers are not similar. Therefore, we set the correlation to be negative and the greater the difference between node degrees, the greater the value. Otherwise, if it is not zero, the value of $\mathcal{NC}$ is the harmonic mean of $\mathcal{NC}_d$ and $\mathcal{NC}_n$. Through the node correlation, we enforce the node vectors closer to the positive larger node correlations among layers and further to the negative node correlations by the following terms:

$$loss_{nc} = \sum_{l_1=1}^{L} \sum_{l_2=l_1+1}^{L} \mid\mid Z^{l_1} - Z^{l_2} \mid\mid \mathcal{NC}^{l_1 l_2}, \tag{18}$$

where Z^{l_1} and Z^{l_2} are the node embedding matrix in G^{l_1} and G^{l_2}, respectively, and $NC^{l_1 l_2}$ is the node correlations matrix between layers l_1 and l_2.

By integrating all objectives, the final objective of the MCME framework can be summarized below:

$$\mathcal{L} = loss_{agg} + loss_{el} + \alpha \cdot loss_{lc} + \beta \cdot loss_{nc}, \tag{19}$$

where α and β are parameters used to control the weight of the regularization terms.

III. Experiment

In this section, we conduct experiments on several real-world multiplex networks to evaluate the performance of the proposed model,. We first introduce the datasets that will be used in the evaluation. Then, we describe the experimental settings. Finally, we show the detailed experimental results.

TABLE I
STATISTICS OF THE DATASETS.

Networks	Layers	Nodes	Edges	Type	$\mathcal{LC}$	$\mathcal{NC}$
CKM	3	246	1,551	Social	0.214	0.062
SacchCere	7	6,570	282,754	Genetic	0.019	-0.488
MUS	7	7,747	1,9842	Genetic	0.010	-0.876
Drosophila	7	8,215	43,366	Genetic	0.005	-0.770
Moscow	3	88,804	210,256	Social	0.040	-0.484

A. Datasets

In our experiments, we work on five public datasets from social and genetic domains. All datasets are obtained from CoMuNe lab's web site[1]. We provide detailed information about each dataset and summarize the dataset statistics in Table I, where $\mathcal{LC}$ and $\mathcal{NC}$ denote the average layer correlations and node correlations among layers of dataset.

B. Baseline Methods

To show the effectiveness of our method, we compare three types of baseline methods, namely, single-layered embedding models, heterogeneous and multiplex embedding models.

- *vgae* [11]: An embedding model for single-layered networks. It is an inference model parameterized by a two-layer GCN.

[1]https://comunelab.fbk.eu/data.php

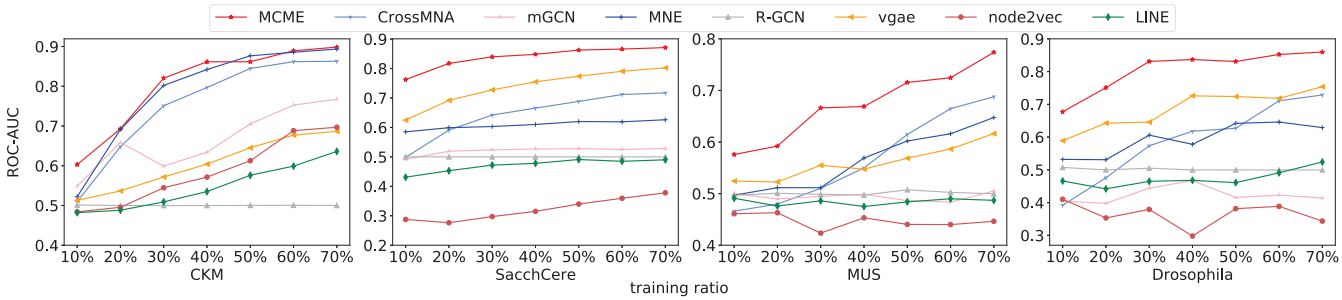

Fig. 3. ROC-AUC scores of link prediction on four datasets.

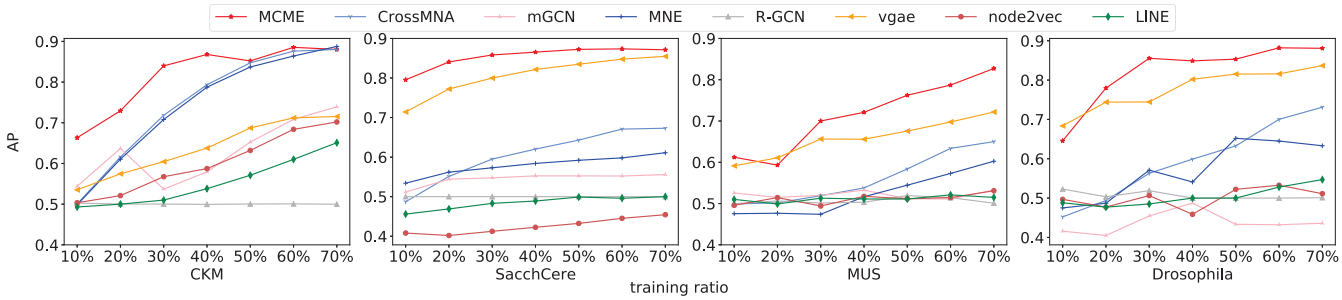

Fig. 4. AP scores of link prediction on four datasets.

- *node2vec* [14]: An embedding model for single-layered networks. It designs a biased random walk and explores diverse neighborhoods to learn richer representations.
- *LINE* [15]: An embedding model for single-layered networks. It preserves both the first-order and second-order proximities in a network.
- *R-GCN* [16]: An embedding model for heterogeneous networks. It develops neural networks to heterogeneous networks and specifically deals with the highly multi-relational data characteristic of realistic knowledge bases.
- *MNE* [3]: An embedding model for multiplex networks. MNE jointly learns a high-dimensional common embedding and a lower-dimensional additional embedding for each layer through a unified embedding model.
- *mGCN* [6]: An embedding model for multiplex networks. mGCN is a multi-dimensional graph convolutional network, which captures the interactions within and across multiple dimensions.
- *CrossMNA* [5]: An embedding model for multiplex networks. The final embedding is generated by combining the layer vector for each layer and the common embedding for each node.

C. Experimental Settings

1) Evaluation Metrics: In our experiments, we perform link prediction tasks at each layer of multiplex networks to verify the effectiveness of learned node embeddings. We randomly split all edges in each layer into two sets for training and testing, respectively. We vary the training set rate from 10% to 70% in 10% increments. We also randomly sample the same number of unconnected node pairs with positive edges in the test set as negative edges and use both the positive and negative edges for testing. Moreover, we use areas under the ROC curve (ROC-AUC) and Average Precision (AP) to evaluate the performance. Higher values of ROC-AUC and AP indicate better link prediction performance.

2) Parameter Settings: The basic experiment we conduct is link prediction in each layer of multiplex networks. So the single-layer network embedding methods learn separately in each layer and then used the learned representation to predict the links in the corresponding network. All models set 128 as the dimension of the final embedding. For LINE, we set the feature embedding dimensions for first-order proximity and second-order proximity both to 64 and concatenate them together as the final node embedding. For CrossMNA, the dimensions of layer vector d_1 and inter-vector d_2 are set as 128 and 100, respectively. For mGCN, we use the representations learned by node2vec on each dataset as input. For MNE, the dimension of additional vectors to be 10 following the original work. Additionally, for node2vec, the best hyper-parameter is empirically set as $p = 2$ and $q = 0.5$.

D. Performance Analysis

We evaluate the quality of the learned embedding by taking link prediction in each layer, and take the average results of all layers as the final result. We evaluate the ROC-AUC and AP values of different models with the training ratio from 10% to 70% in 10% increments. We take five times experiments for each training ratio and take the average as the final result. The results are shown in Figs. 3 and 4 and Table II, where Fig. 3 shows the ROC-AUC scores on four relatively small datasets, Fig. 4 shows the AP scores and Table II shows the ROC-AUC and AP scores on Moscow and the test error (mean absolute error) on this dataset. Note that due to out of memory, we do

TABLE II
THE ROC-AUC AND AP SCORES OF LINK PREDICTION ON THE MOSCOW DATASET, WHERE 10%-40% INDICATES THE TRAINING RATE AND THE STANDARD DEVIATIONS ARE REPORTED IN THE PARENTHESES.(BOLD IS BEST)

Model	10%		20%		30%		40%	
	ROC-AUC	AP	ROC-AUC	AP	ROC-AUC	AP	ROC-AUC	AP
node2vec	0.425(0.000)	0.472(0.000)	0.393(0.002)	0.458(0.001)	0.376(0.001)	0.451(0.001)	0.371(0.001)	0.451(0.001)
LINE	0.496(0.004)	0.498(0.002)	0.490(0.008)	0.496(0.003)	0.500(0.001)	0.500(0.002)	0.501(0.002)	0.501(0.002)
R-GCN	0.500(0.000)	0.500(0.000)	0.500(0.001)	0.499(0.000)	0.375(0.001)	0.455(0.001)	0.501(0.000)	0.500(0.000)
MNE	0.541(0.001)	0.501(0.000)	0.575(0.000)	0.505(0.001)	0.589(0.001)	0.501(0.528)	0.0.627(0.002)	0.551(0.002)
mGCN	0.500(0.001)	0.523(0.000)	0.505(0.001)	0.530(0.001)	0.510(0.001)	0.533(0.006)	0.511(0.002)	0.537(0.001)
CrossMNA	0.467(0.001)	0.484(0.001)	0.547(0.002)	0.531(0.001)	0.616(0.002)	0.581(0.001)	0.674(0.002)	0.629(0.003)
MCME	**0.624**(0.004)	**0.717**(0.004)	**0.684**(0.001)	**0.756**(0.002)	**0.713**(0.001)	**0.773**(0.001)	**0.721**(0.002)	**0.776**(0.002)

not experiment with them on Moscow with vgae. From the experiment results, we can make the following observations:

- Our proposed model MCME almost outperforms the other baselines on all datasets. Especially, MCME can yield significant improvement on sparse, small layer correlation and negative node correlation networks. For example, on the MUS dataset by the ROC-AUC, MCME can boost the performance around 5% -11% compared to the best baseline when the training rate is increased from 10% to 70%.
- As the rate of training data gradually increases, the results of all models will be improved. That's because as the training rate increases, the network structure becomes more complete and the information becomes more adequate.
- Our model can achieve promising results with a small rate of the training set. Almost on all the datasets by the 30% of training rate can achieve comparable results with 70% but on the MUS dataset because it is too sparse.
- In general, multiplex embedding models can obtain better results than single-layered embedding models, especially on the CKM and Moscow, where network structures in each layer are relatively similar. This indicates that modeling the complex dependencies across the different layers is effective.

IV. CONCLUSION

Multiplex networks are becoming an increasingly common practice in real life, in which multiple types of connectivity relations exist among a set of nodes. Most existing studies work overlooked the complex properties of correlations in multiplex networks. In this paper, we propose MCME, a novel embedding method for multiplex networks, which models the complex correlations in multiplex networks. We test our method for link prediction tasks using five datasets compared with some state-of-the-art baseline models. The experimental results show the generalization ability of our model. It obtains comparable performance in the aspect of connectivity consistency after multiplexing, and has significant advantages on the multiplex networks with different structures. In future work, we will extend our model to weighted and heterogeneous networks.

REFERENCES

[1] D. Greene and P. Cunningham, "A matrix factorization approach for integrating multiple data views," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 5781. Springer, 2009, pp. 423–438.

[2] J. Liu, C. Wang, J. Gao, and J. Han, *Multi-view clustering via joint nonnegative matrix factorization*, 2013.

[3] H. Zhang, L. Qiu, L. Yi, and Y. Song, "Scalable multiplex network embedding." in *IJCAI*, vol. 18, 2018, pp. 3082–3088.

[4] S. K. Ata, Y. Fang, M. Wu, J. Shi, C. K. Kwoh, and X. Li, "Multi-view collaborative network embedding," *arXiv preprint arXiv:2005.08189*, 2020.

[5] X. Chu, X. Fan, D. Yao, Z. Zhu, J. Huang, and J. Bi, "Cross-network embedding for multi-network alignment," in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, and L. Zia, Eds. ACM, 2019, pp. 273–284.

[6] Y. Ma, S. Wang, C. C. Aggarwal, D. Yin, and J. Tang, "Multi-dimensional graph convolutional networks," in *Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, Alberta, Canada, May 2-4, 2019*. SIAM, 2019, pp. 657–665.

[7] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, and B. Wang, *One2Multi Graph Autoencoder for Multi-View Graph Clustering*. New York, NY, USA: Association for Computing Machinery, 2020, p. 3070–3076.

[8] C. Park, D. Kim, J. Han, and H. Yu, "Unsupervised attributed multiplex network embedding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 5371–5378, 2020.

[9] J. Li, C. Chen, H. Tong, and H. Liu, *Multi-Layered Network Embedding*. Proceedings of the 2018 SIAM International Conference on Data Mining, 2018.

[10] J. Coleman and K. H. Menzel, "The diffusion of an innovation among physicians," *Social Networks*, vol. 20, no. 4, pp. 253–270, 1957.

[11] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *stat*, vol. 1050, p. 21, 2016.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2013, pp. 3111–3119.

[14] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM, 2016, pp. 855–864.

[15] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*. ACM, 2015, pp. 1067–1077.

[16] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. V. Berg, and M. Welling, *Modeling Relational Data with Graph Convolutional Networks*. The Semantic Web, 2018.