

# Multi-View Collaborative Network Embedding

SEZIN KIRCALI ATA\*, Nanyang Technological University, Singapore

YUAN FANG<sup>†</sup>, Singapore Management University, Singapore

MIN WU<sup>‡</sup>, Institute for Infocomm Research, Singapore

JIAQI SHI<sup>‡</sup>, Singapore Management University

CHEE KEONG KWOH, Nanyang Technological University

XIAOLI LI, Institute for Infocomm Research and Nanyang Technological University

Real-world networks often exist with multiple views, where each view describes one type of interaction among a common set of nodes. For example, on a video-sharing network, while two user nodes are linked if they have common favorite videos in one view, they can also be linked in another view if they share common subscribers. Unlike traditional single-view networks, multiple views maintain different semantics to complement each other. In this paper, we propose MANE, a multi-view network embedding approach to learn low-dimensional representations. Similar to existing studies, MANE hinges on diversity and collaboration—while diversity enables views to maintain their individual semantics, collaboration enables views to work together. However, we also discover a novel form of *second-order* collaboration that has not been explored previously, and further unify it into our framework to attain superior node representations. Furthermore, as each view often has varying importance *w.r.t.* different nodes, we propose MANE<sup>+</sup>, an *attention*-based extension of MANE to model node-wise view importance. Finally, we conduct comprehensive experiments on three public, real-world multi-view networks, and the results demonstrate that our models consistently outperform state-of-the-art approaches.

CCS Concepts: • **Information systems** → **Data mining**; *Social networks*; • **Computing methodologies** → **Learning latent representations**.

Additional Key Words and Phrases: multi-view networks, network embedding

## ACM Reference Format:

Sezin Kircali Ata, Yuan Fang, Min Wu, Jiaqi Shi, Chee Keong Kwoh, and Xiaoli Li. 2020. Multi-View Collaborative Network Embedding. *ACM Trans. Knowl. Discov. Data.* 1, 1, Article 1 (January 2020), 17 pages. <https://doi.org/10.1145/3441450>

\*The author is currently affiliated with KK Women's and Children's Hospital, Singapore.

<sup>†</sup>Both authors are the corresponding authors.

<sup>‡</sup>The author is currently affiliated with University of California Irvine, United States.

Authors' addresses: Sezin Kircali Ata, Nanyang Technological University, 50 Nanyang Ave, Singapore, 639798, [sezin001@e.ntu.edu.sg](mailto:sezin001@e.ntu.edu.sg); Yuan Fang, Singapore Management University, 81 Victoria St, Singapore, 188065, [yfang@smu.edu.sg](mailto:yfang@smu.edu.sg); Min Wu, Institute for Infocomm Research, 1 Fusionopolis Way, Singapore, 138632, [wumin@i2r.a-star.edu.sg](mailto:wumin@i2r.a-star.edu.sg); Jiaqi Shi, Singapore Management University, [jqshi@smu.edu.sg](mailto:jqshi@smu.edu.sg); Chee Keong Kwoh, Nanyang Technological University, [asckkwoh@ntu.edu.sg](mailto:asckkwoh@ntu.edu.sg); Xiaoli Li, Institute for Infocomm Research and Nanyang Technological University, [xlli@i2r.a-star.edu.sg](mailto:xlli@i2r.a-star.edu.sg).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2020 Association for Computing Machinery.

1556-4681/2020/1-ART1 \$15.00

<https://doi.org/10.1145/3441450>

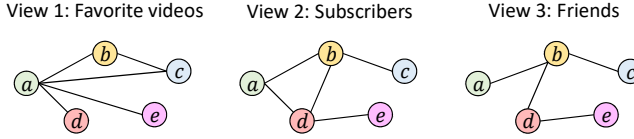


Fig. 1. A toy video-sharing network with three views.

## 1 INTRODUCTION

Large-scale network data are ubiquitous in many domains, including social networks, biological networks and transportation networks. Practical applications on these networks, such as personalized recommendations [44] and disease protein predictions [1], can often be cast as link prediction and node classification tasks. Naturally, an important step towards solving these problems is to derive effective representations from the networks. In particular, network embedding has emerged as a promising direction and achieved considerable success, which aims to learn a low-dimensional and continuous vector representation for each node on the network.

While many prior studies [27, 32] deal with single-view networks, multiple views naturally arise in real-world scenarios. They are known as *multi-view* or *multiplex* networks [28, 41], as shown in Fig. 1. The toy example comprises three views: the same set of users can be linked in three different ways, through sharing common favorite videos, subscribers or friends on a video-sharing platform. Since a single view is sometimes sparse and noisy, it becomes beneficial to exploit multiple views jointly for learning more effective and robust representations.

**Prior work.** In this paper, we study the problem of multi-view network embedding. Although numerous multi-view learning algorithms exist [39] in various contexts such as classification [3, 19] and clustering [6], fewer research have investigated multi-view network embedding [28, 41]. Existing multi-view network embedding approaches typically leverage two common *characteristics* on multi-view data, namely, *diversity* and *collaboration*. In terms of diversity, as each view is constructed from different semantics, view-specific representations should be learned for each view in order to retain the diverse semantics of various views. For instance, in Fig. 2(a), users *a* and *c* are connected in view 1 as they share common favorite videos, but they have no common subscribers in view 2. Thus, *a* and *c*'s representations should be close to one another in view 1 but not necessarily close in view 2. This can be addressed by learning one set of node representations in each view. In terms of collaboration, as the same node in different views ultimately describes the same instance (e.g., a particular user on a social network), its behaviors across views are not completely independent. Thus, for the same node, its view-specific representations for different views should be synergistic and collaborate with each other, as illustrated by the cross-view alignment in Fig. 2(b).

**Our insight.** While the above form of collaboration has been adopted in several state-of-the-art approaches [28, 29, 41], it does not adequately address the collaboration of multiple views. Specifically, for every node, its representations across views are aligned (*i.e.*, made close) indiscriminately, regardless of its associations (or the lack thereof) with other nodes. However, such associations are often important to multi-view collaboration—two nodes associated in one view have a better chance to also associate with each other in a different view. Our analysis of three real-world multi-view networks reveals that, two nodes linked in one view are 2.5–27.8 times more likely to also form links in a second view than two nodes not linked in the first view.<sup>1</sup> In other words, two nodes associated in at least one view are more likely to lead to multi-view collaboration. Taking Fig. 1

<sup>1</sup>The three multi-view networks will be elaborated in our experiments in Sect. 5, and more details of the analysis will be presented in Sect. 4.

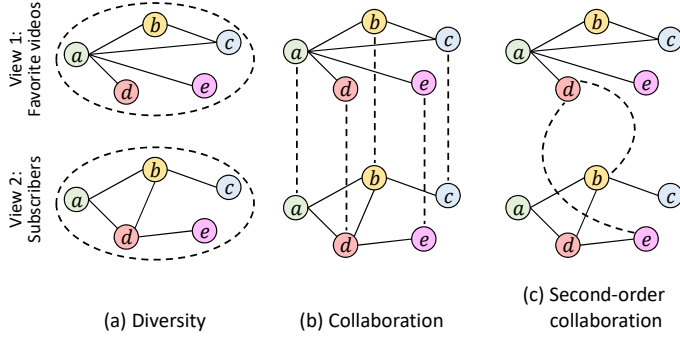


Fig. 2. Three characteristics of a multi-view network.

as an example, users  $d$  and  $e$  have common subscribers in view 2, implying that they are likely to also “favorite” common videos in future, even though they are currently not linked in view 1. Thus, it is reasonable to pay more attention to  $d$  and  $e$  for multi-view collaboration. On the other hand,  $b$  and  $e$  are not linked in any view, which are therefore less crucial to collaboration. We call this phenomenon the *second-order* collaboration, since it deals with a pair of nodes across views. In contrast, the collaboration shown in Fig. 2(b) is first-order, as it only deals with individual nodes.

**Present work.** We aim to capture a novel form of second-order collaboration, the third characteristic of multi-view networks in addition to the previously studied diversity and first-order collaboration. To achieve the first-order collaboration for a given node, say  $d$ , its view-specific representations are aligned across all views, as illustrated in Fig. 2(b). To model the second-order collaboration for  $d$ , we further take account of  $d$ ’s associations with other nodes. Using the earlier example,  $d$ ’s representation in view 1 should exploit the knowledge that  $d$  is linked to  $b$  and  $e$  in view 2. Unlike the first order which directly moves  $d$ ’s representation in view 1 towards its representation in view 2, we move it towards the representations of  $b$  and  $e$  in view 2 to achieve the second-order multi-view collaboration, as illustrated in Fig. 2(c). The advantage of this design is twofold. First, it explicitly exploits the associations between nodes for multi-view collaboration, which carry valuable information. Second, it results in node-wise differentiation, since the extent of collaboration a node can leverage in one view depends on its associations in other views. Both elements are neglected by the first-order collaboration in existing work.

In this paper, we unify the three characteristics of a multi-view network into one framework, and propose a new unsupervised algorithm for Multi-view collaborative Network Embedding (MANE). Furthermore, in a multi-view network, not all views are equally important. However, their importance to the nodes is often *non-uniform*, varying from node to node. Inspired by the neural attention mechanism [2], we develop an approach called MANE<sup>+</sup>, which is an extension of MANE to consider node-wise view importance. The attention-based approach enables each node to locate and focus on its most important and relevant view.

We summarize the main contributions as follows. (i) We discover a new form of second-order collaboration on multi-view networks. (ii) We propose a novel multi-view network embedding algorithm MANE, integrating different characteristics in a unifying framework. (iii) We further develop MANE<sup>+</sup> to capture node-wise view importance through an attention mechanism. (iv) We conduct extensive experiments on three public datasets, and empirically demonstrate the superiority of our approaches.

## 2 RELATED WORK

Network embedding has been extensively studied for its importance in real-world applications. Earlier studies [27, 32] have primarily devoted to preserving network structures through neighborhood sampling, such as random walks and first- or second-order proximity. More recent algorithms adopt different paradigms towards the same goal, such as GraphGAN [35] based on generative adversarial nets, and EP [12] based on message passing on the network. Meanwhile, graph neural networks [17, 34] have recently emerged as competitive end-to-end approaches.

While the above algorithms have achieved promising results, they are only designed for single-view networks, as opposed to multi-view networks investigated in this paper. In multi-view networks, each view describes one form of interaction among a common set of nodes. This is analogous to traditional multi-view data, where each view forms a distinct feature set for a common set of instances [39], which have been studied in various contexts [3, 6, 9, 13, 19, 21, 23]. Learning on multi-view networks have also been studied, such as spectral clustering [18, 22, 24] and classification [14, 38].

Recent approaches such as MVE [28], MVNE [30], mvn2vec [29] and MNE [41] intend to be more general, which aim to learn low-dimensional representations that preserve the structures of multiple views. The learned representations are often generic enough to support various applications such as node classification and link prediction. However, in addition to the characteristic of diversity, these multi-view network embedding methods only account for the first-order collaboration to align the representations of each node across views. In our approach, we discover and leverage the second-order collaboration, which promotes cross-view reinforcement based on how nodes associate with others. Most notably, Ni *et al.* [25] have proposed the concept of proximity disagreement, which requires the instances of the same node across views to have similar proximity to all other nodes. While this is a weaker assumption than the first-order collaboration, it is still different from our second-order collaboration as it does not leverage valuable node associations for fine-grained collaboration that varies from node to node. Moreover, a few attention-based algorithms exist [28, 37] to differentiate the importance of views in multi-view networks, and a more general definition of multi-view network has also been explored to model many-to-many node mappings across views [25]. In a related but different formulation, network structures have also been studied in conjunction with content features, where the network structure can be treated as one view, and node contents as another view [16, 40, 42].

On another line, network embedding algorithms have also been designed for heterogeneous information networks (HIN) [31]. A HIN consists of multiple types of nodes and edges, carrying different semantics similar to multiple views. However, HIN embedding [5, 10, 11, 15, 43, 45] typically fuses and merges different types into only one view, and only one set of node embeddings are learnt without view-specific representations. Thus, they are fundamentally single-view models, and do not work well empirically on multi-view networks, as demonstrated in our experiments. A recent study GATNE [4] employs a more general form of MNE [41] to further explore representation learning on multi-view HINs. Again, GATNE does not consider the second-order collaboration.

## 3 PRELIMINARIES

We begin with a formal definition of multi-view network, followed by the problem formulation. The main notations are listed in Table 1.

**DEFINITION 1.** *A multi-view network consists of a set of nodes  $U$  and a set of views  $V$ , where each view  $v \in V$  contains a set of edges  $E^{(v)}$  between the nodes in  $U$ . Specifically, an edge  $e_{ij}^{(v)} \in E^{(v)}$  links nodes  $i, j \in U$  in view  $v \in V$ . We denote this multi-view network as  $G = (U, V, \{E^{(v)} : v \in V\})$ .*

Table 1. Summary of main notations.

Notation	Definition
$U$	the set of nodes $U$
$V$	the set of views $V$
$E^{(v)}$	the set of edges $E^{(v)}$ in view $v$
$\mathbf{f}_i^{(v)}, \tilde{\mathbf{f}}_i^{(v)}$	center and context embeddings of node $i$ for view $v$
$\mathbf{f}_i$	final (output) embedding of node $i$
$\Omega^{(v)}$	the set of intra-view pairs in view $v$
$i^{(v)}$	the instance of node $i$ in view $v$
$a_i^{(v)}$	attention-based weight of view $v$ w.r.t. node $i$
$\mathbf{f}_i^a$	attention-based embedding of node $i$
$\alpha, \beta, \gamma$	hyperparameters to balance loss components

Given such a network  $G$ , the goal of multi-view network embedding is to learn a low-dimensional vector representation  $\mathbf{f}_i \in \mathbb{R}^D$  for each node  $i \in U$ , where  $D$  is the dimension of the embedding space. More specifically, to maintain the diversity of views, for each node  $i \in U$  and each view  $v \in V$ , we learn an intermediate view-specific representation  $\mathbf{f}_i^{(v)} \in \mathbb{R}^{\lfloor D/|V| \rfloor}$ . The final representation  $\mathbf{f}_i$  is aggregated from these intermediate representations:

$$\mathbf{f}_i = \oplus_{v \in V} \mathbf{f}_i^{(v)}, \quad (1)$$

where  $\oplus$  denotes some form of aggregation, such as vector concatenation as adopted in our formulation. The final representations can be used as features in a variety of applications such as node classification and visualization, relationship mining and link prediction.

## 4 PROPOSED APPROACH

In this section, we introduce our models MANE and MANE<sup>+</sup> for multi-view network embedding.

### 4.1 Overall Framework

To start, we present the overall framework in Fig. 3. The three characteristics of a multi-view network can be unified by three categories of node pairs.

In the first category, we consider *intra-view* pairs, which can be generated using random walks in each view [27]. By modeling intra-view pairs in each view, we are able to capture the diversity of different views. In the second category, we consider *cross-view, intra-node* pairs, where instances of the same node (*i.e.*, intra-node) across two views (*i.e.*, cross-view) form pairs. By aligning node representations in such a pair, the first-order collaboration can be captured. In the third category, we consider *cross-view, cross-node* pairs, where a node in one view form pairs with different nodes (*i.e.*, cross-node) in another view (*i.e.*, cross-view) based on their associations in each view, to capture the second-order collaboration.

The three categories of node pairs are jointly trained to derive one set of embeddings for each view, which are further aggregated to produce the final embeddings. In particular, during aggregation, MANE assumes equal weight across views, and MANE<sup>+</sup> learns a different weight for each view.

### 4.2 Characteristics of Multi-View Networks

In the following, we delve into each characteristic and its corresponding node pairs.

**Diversity: Intra-view pairs.** To retain the diversity of each view, each node has a view-specific representation such that two nodes associated in the same view should have similar representations

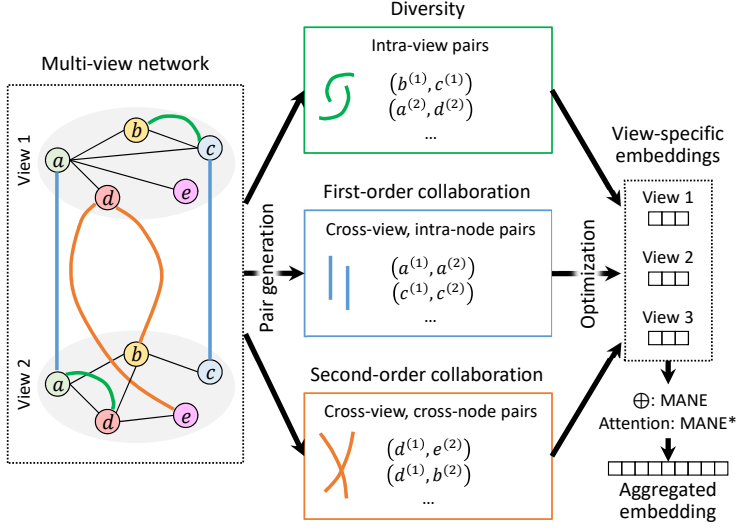


Fig. 3. Overall framework of MANE.

for that view. Following the DeepWalk model [27], for each view, we generate topologically associated node pairs from random walks. Such pairs are also intra-view, as each pair is generated within one view. As shown in Fig. 3, we consider intra-view pairs such as  $(b^{(1)}, c^{(1)})$  in view 1 and  $(a^{(2)}, d^{(2)})$  in view 2, where the superscripts indicate the views from which the node instances are observed.

Formally, for a given view  $v \in V$ , we assume a set of intra-view pairs  $\Omega^{(v)} \subset U \times U$ . Adopting the well-known skipgram model [27], a pair  $(i^{(v)}, j^{(v)}) \in \Omega^{(v)}$  consists of a center node  $i^{(v)}$  and a context node  $j^{(v)}$ . Given a center node, the task is to predict its context node, i.e., to maximize  $P(j^{(v)}|i^{(v)})$ . Subsequently, we optimize the following loss for some model parameters  $\Theta$ :

$$L_{\text{Div}}(\Theta) = - \sum_{v \in V} \sum_{(i^{(v)}, j^{(v)}) \in \Omega^{(v)}} \log P(j^{(v)}|i^{(v)}; \Theta). \quad (2)$$

Here  $P(j^{(v)}|i^{(v)}; \Theta)$  is further defined with a softmax function:

$$P(j^{(v)}|i^{(v)}; \Theta) = \frac{\exp(\tilde{\mathbf{f}}_j^{(v)} \cdot \mathbf{f}_i^{(v)})}{\sum_{u \in U} \exp(\tilde{\mathbf{f}}_u^{(v)} \cdot \mathbf{f}_i^{(v)})}, \quad (3)$$

where  $\mathbf{f}_i^{(v)}, \tilde{\mathbf{f}}_i^{(v)} \in \mathbb{R}^{[D/|V|]}$  respectively denote the center and context embeddings of node  $i$  specific to view  $v$ . Thus, the model parameters consist of the center and context embeddings of all nodes in all views, i.e.,  $\Theta = \{\mathbf{f}_i^{(v)}, \tilde{\mathbf{f}}_i^{(v)} : (i, v) \in U \times V\}$ . Essentially, the views are decoupled from each other to capture the diversity of each view, as no cross-view pairs are involved.

**First-order collaboration: Cross-view, intra-node pairs.** While different views exhibit diversity, they ultimately converge on a common set of nodes. Since the instances of the same node across different views fundamentally describes the same entity, such as a particular user on a social network, its view-specific representations should collaborate with one another. As shown in Fig. 3, we consider cross-view, intra-node pairs such as  $(a^{(1)}, a^{(2)})$  and  $(c^{(1)}, c^{(2)})$ , which can be generated across different views for the same node. In each pair, the two view-specific embeddings of the

same node should become similar, by optimizing the following loss:

$$\begin{aligned}
 L_{C1}(\Theta) &= - \sum_{v \in V} \sum_{(i^{(v)}, \cdot) \in \Omega^{(v)}} \sum_{v' \neq v} \log P(i^{(v')} | i^{(v)}; \Theta) \\
 &= - \sum_{v \in V} \sum_{(i^{(v)}, \cdot) \in \Omega^{(v)}} \sum_{v' \neq v} \log \frac{\exp(\mathbf{f}_i^{(v')} \cdot \mathbf{f}_i^{(v)})}{\sum_{u \in U} \exp(\mathbf{f}_u^{(v')} \cdot \mathbf{f}_i^{(v)})}.
 \end{aligned} \tag{4}$$

Note that we form a cross-view, intra-node pair  $(i^{(v)}, i^{(v')})$  for the center node  $i$  in every intra-view pair  $(i^{(v)}, \cdot) \in \Omega^{(v)}$ . Thus, a frequently observed center node in a view, which has more topological importance in that view, will also be updated more in the first-order collaboration to ensure a balanced optimization.

**Second-order collaboration: Cross-view, cross-node pairs.** As motivated in Sect. 1, our key intuition is that two nodes linked in one view are often more likely to also link with each other in another view. Given two views  $v \neq v'$ , this intuition can be quantified by the following ratio:

$$\frac{P(e_{ij}^{(v')} \in E^{(v')} | e_{ij}^{(v)} \in E^{(v)})}{P(e_{ij}^{(v')} \in E^{(v')} | e_{ij}^{(v)} \notin E^{(v)})}. \tag{5}$$

The larger the ratio, the more likely that two nodes linked in one view will also form a link with each other in another view; a ratio of 1 would imply that whether two nodes are linked in one view has no bearing on their link formation in another view. On the three real-world multi-view networks used in our experiments, namely Alzheimer's, LinkedIn and YouTube (details in Sect. 5), the above ratio for all pairwise views ranges from 2.5 to 27.8 with a median of 7.0 as shown in Table 2. That means on these networks, node pairs linked in one view are significantly more likely than those not linked to also form links in another view. Thus, the second-order collaboration is a fair assumption on many multi-view networks, although it is not universal. As we shall see later, our model is flexible to adjust the contribution from the second-order collaboration.

Using Fig. 3 as an example, nodes  $d$  and  $e$  are topologically related in view 2 but not in view 1. Under the second-order collaboration, the two nodes may entail a latent, unknown association in view 1, which is not currently manifested, possibly due to incomplete and noisy data, or a time delay—they may develop an explicit link in future. It is important to note that this assumption merely suggests a tendency. Thus, it is not appropriate to directly or explicitly make  $d$  and  $e$ 's representations closer in view 1. Instead, we propose an implicit mechanism to enable such collaboration across views. As shown in Fig. 3, we consider cross-view, cross-node pairs like  $(d^{(1)}, e^{(2)})$  and  $(d^{(1)}, b^{(2)})$ . That is, we move  $d$ 's representation in view 1 towards  $e$  and  $b$ 's in view 2, enabling the two views to collaborate depending on  $d$ 's associations to other nodes. In contrast, the first-order collaboration simply make  $d$ 's representations in both views closer without the knowledge of  $d$ 's associations.

Formally, suppose there exists an intra-view pair in view  $v$ , say  $(i^{(v)}, j^{(v)}) \in \Omega^{(v)}$ , which means node  $i$  is topologically associated with  $j$  in view  $v$ . Given such an association, we can form a cross-view, cross-node pair  $(i^{(v')}, j^{(v)})$ , to move  $i$ 's representations in another view  $v'$  towards  $j$ 's

Table 2. Second-order collaboration between pairwise views ( $v, v'$ ) in three multi-view networks.

$v \setminus v'$	GO	PPI
GO	—	27.8
PPI	27.7	—

(a) Alzheimer's

$v \setminus v'$	College	Employer	Friend
College	—	6.4	27.8
Employer	6.6	—	12.6
Friend	15.7	9.1	—

(b) LinkedIn

$v \setminus v'$	Friend	Subscriber	Video
Friend	—	6.6	4.5
Subscriber	7.5	—	2.5
Video	5.1	2.6	—

(c) YouTube

representation in view  $v$ . Thus, we optimize the following loss:

$$\begin{aligned}
 L_{C2}(\Theta) &= - \sum_{v \in V} \sum_{(i^{(v)}, j^{(v)}) \in \Omega^{(v)}} \sum_{v' \neq v} \log P(i^{(v')}, j^{(v)}; \Theta) \\
 &= - \sum_{v \in V} \sum_{(i^{(v)}, j^{(v)}) \in \Omega^{(v)}} \sum_{v' \neq v} \frac{\exp(\tilde{\mathbf{f}}_j^{(v)} \cdot \mathbf{f}_i^{(v')})}{\sum_{u \in U} \exp(\tilde{\mathbf{f}}_u^{(v)} \cdot \mathbf{f}_i^{(v')})}.
 \end{aligned} \tag{6}$$

### 4.3 The MANE Algorithm

With the losses based on the three categories of node pairs, we further present the proposed approach MANE.

**Loss function.** We combine the losses of the three characteristics in Eq. (2), (4) and (6) into the following, as the overall loss for a multi-view network.

$$L = L_{\text{Div}} + \alpha \cdot L_{C1} + \beta \cdot L_{C2}, \tag{7}$$

where  $\alpha, \beta \geq 0$  are hyperparameters to control the relative importance among the three components. This is an unsupervised formulation, where we maximize the likelihood of the parameters  $\Theta$  only based on the three categories of node pairs, and produce final embeddings as features for downstream applications. However, depending on the application, supervision may still be leveraged, such as node class labels in node classification.

As a further remark, when  $\alpha = \beta = 0$ , MANE reduces to Skip-gram models such as DeepWalk [27] on individual views in a decoupled manner, without leveraging any collaboration between views. When  $\alpha > 0$  and  $\beta = 0$ , MANE only captures diversity and first-order collaboration.

**Algorithm.** Our approach is summarized in Algorithm 1. We first initialize the parameters randomly (line 1). Next, for each view  $v$ , we sample random walks starting from every node, and



**Algorithm 1:** Training of MANE

---

**Input** :  $G = (U, V, \{E^{(v)} : v \in V\})$ ,  $\alpha, \beta$   
**Output**: final embeddings  $\{\mathbf{f}_i : i \in U\}$

- 1 Randomly initialize parameters  $\Theta$
- 2 **for** each view  $v \in V$  **do**
- 3      $R^{(v)} = \text{SampleRandomWalks}(U, E^{(v)})$
- 4      $\Omega^{(v)} = \text{GenerateIntraViewPairs}(R^{(v)})$
- 5 **while** not converged **do**
- 6     **for** each view  $v \in V$  **do**
- 7         Shuffle pairs in  $\Omega^{(v)}$
- 8         **for** each intra-view pair  $(i^{(v)}, j^{(v)}) \in \Omega^{(v)}$  **do**
- 9             Update  $\mathbf{f}_i^{(v)}, \tilde{\mathbf{f}}_j^{(v)}, \tilde{\mathbf{f}}_u^{(v)}$  based on Eq. (2)
- 10         **for** each view  $v' \neq v$  **do**
- 11             Form a cross-view, intra-node pair  $(i^{(v)}, i^{(v')})$
- 12             Update  $\mathbf{f}_i^{(v)}, \mathbf{f}_i^{(v')}, \mathbf{f}_u^{(v')}$  based on Eq. (4)
- 13             Form a cross-view, cross-node pair  $(i^{(v')}, j^{(v)})$
- 14             Update  $\mathbf{f}_i^{(v')}, \tilde{\mathbf{f}}_j^{(v)}, \tilde{\mathbf{f}}_u^{(v)}$  based on Eq. (6)
- 15 **for**  $i \in U$  **do**
- 16      $\mathbf{f}_i = \oplus_{v \in V} \mathbf{f}_i^{(v)}$
- 17 **return**  $\{\mathbf{f}_i : i \in U\}$

---

further generate intra-view node pairs  $\Omega^{(v)}$  based on the sampled random walks (lines 2–4). In the succeeding block, we perform gradient updates on the parameters (lines 5–14). More specifically, we form intra-view node pairs to capture diversity (lines 8–9), cross-view, intra-node pairs to capture first-order collaboration (lines 11–12), and cross-view, cross-node pairs to capture second-order collaboration (lines 13–14). Finally, we concatenate the view-specific center embeddings of each node to obtain the final embedding (lines 15–16).

To implement the algorithm, we follow DeepWalk [27] to perform truncated random walks in each view, as well as to further generate center-context node pairs from the sampled random walks. Such center-context pairs form the intra-view pairs. We also adopt negative sampling for the softmax function in each loss component to accelerate the training process. To minimize the overall loss function, we employ the Adam optimizer.

**Complexity analysis.** Based on existing work [28], learning view-specific embeddings over all views takes  $O(|E| \cdot D/|V| \cdot K)$  time, where  $|E| = \sum_{v \in V} |E^{(v)}|$  denotes the total number of edges in all views,  $\lfloor D/|V| \rfloor$  is the dimension for view-specific embeddings, and  $K$  is the number of negative samples. This is equivalent to the time complexity of intra-view-cross-node consistency. On the other hand, the complexities for cross-view-intra-node and cross-view-cross-node consistencies are both  $O(|E| \cdot D/|V| \cdot K \cdot |V|)$ . Thus, the overall time complexity is  $O(|E| \cdot D \cdot K)$ . Thus, MANE scales linearly in the total number of edges in all views.

#### 4.4 Extension with View Attention

In MANE, we treat all views with equal importance during the final aggregation. However, in real-world scenarios, the importance or relevance of each view is often non-uniform, and varies

from node to node. Thus, we propose MANE<sup>+</sup>, an extension of MANE to account for node-wise view importance.

**View attention.** To allow a node to focus on the view most important to it, we adopt the neural attention mechanism [2] to model view attention. We first define a score function  $s : U \times V \rightarrow \mathbb{R}$  to compute the similarity between any node  $i$  and any view  $v$ :

$$s(i, v) = \mathbf{z}_2^{(v)} \cdot \tanh(\mathbf{z}_1^{(v)} \circ \mathbf{f}_i + b_1^{(v)} \mathbf{1}) + b_2^{(v)}, \quad (8)$$

where  $\mathbf{z}_1^{(v)}, \mathbf{z}_2^{(v)} \in \mathbb{R}^D, b_1^{(v)}, b_2^{(v)} \in \mathbb{R}$  are trainable parameters specific to view  $v$ , and  $\circ$  denotes the Hadamard product. The scores are further normalized to obtain the attention value  $a_i^{(v)}$ , i.e., the weight of view  $v$  w.r.t. node  $i$ :

$$a_i^{(v)} = \frac{\exp s(i, v)}{\sum_{v' \in V} \exp s(i, v')}. \quad (9)$$

Subsequently, for every node  $i$ , we aggregate its view-specific embeddings into a vector  $\mathbf{f}_i^a \in \mathbb{R}^D$ , weighted by its view attention:

$$\mathbf{f}_i^a = \tanh \left( \oplus_{v \in V} a_i^{(v)} \mathbf{f}_i^{(v)} \right). \quad (10)$$

The attention-based embeddings,  $\{\mathbf{f}_i^a : i \in U\}$ , can be used as features for downstream applications, where some supervision is often available to guide the learning of view-specific attention parameters [28]. For example, we could minimize the following loss for node classification:

$$L_{\text{Att}} = \frac{1}{|U_{\text{train}}|} \sum_{i \in U_{\text{train}}} \text{CrossEntropy}(y_i, h(\mathbf{f}_i^a)), \quad (11)$$

where  $U_{\text{train}}$  is the set of training nodes,  $y_i$  is the class of node  $i$ , and  $h(\mathbf{f}_i^a)$  is the classification output for node  $i$  using the attention-based embedding  $\mathbf{f}_i^a$ . Since  $U_{\text{train}} \subset U$ , this is a semi-supervised setting. Similarly, for link prediction an alternative classification function  $h'(\mathbf{f}_i^a, \mathbf{f}_j^a)$  could be employed, using the attention-based embedding of two nodes.

**Overall loss.** We formulate the loss of MANE<sup>+</sup> by extending the loss of MANE in Eq. (7), to include the attention component:

$$L^+ = L_{\text{Div}} + \alpha \cdot L_{C1} + \beta \cdot L_{C2} + \gamma \cdot L_{\text{Att}}, \quad (12)$$

where  $\gamma > 0$  is a hyperparameter to control the contribution of the attention-based loss.

Due to the additional loss component, compared to MANE, an extra time complexity of  $O(T \cdot D \cdot |V|)$  will be added to MANE<sup>+</sup>, where  $T$  is the size of the training data. Since MANE is linear in the number of edges  $|E|$ , and  $T$  is typically much smaller than  $|E|$  for node classification and comparable in scale to  $|E|$  for link prediction, the extra complexity term does not present a significant overhead.

## 5 EXPERIMENTS

We evaluate the performance of our proposed methods and conduct an in-depth model analysis on three real-world multi-view networks. Results demonstrate the superior performance of MANE and MANE<sup>+</sup> against a comprehensive suite of state-of-the-art baselines.

### 5.1 Experimental Setup

**Datasets and tasks.** Table 3 summarizes three public, real-world multi-view networks used in our experiments, as follows.

**Alzheimer's:** A protein network to identify causative genes for Alzheimer's disease, which includes two views, namely, PPI and GO. The PPI view captures the protein-protein interactions

Table 3. Summary of datasets.

Dataset	# Nodes	# Edges in each view		
		PPI	GO	
Alzheimer's	12 901	96 845	107 508	
LinkedIn	10 196	College	Employer	Friend
		1 527 681	1 765 927	29 434
YouTube	7 558	Friend	Subscriber	Video
		1 003 923	1 547 091	1 918 208

from the IntAct database [26], whereas the GO view captures the functional associations between proteins based on their Gene Ontology (GO) annotations [36]. For the GO view, we first calculated the pairwise functional similarity between proteins, and subsequently built a K-NN similarity graph with  $K = 10$ . In this two-view protein network, each node has a binary label to show whether it is a causative gene for Alzheimer's disease. In other words, the task of disease gene identification for Alzheimer's can be cast as a binary classification problem. In particular, this is a highly skewed dataset with fewer than 1% positives. Thus, we have also tried to over-sample with SMOTE [7], although it yielded similar results.

**LinkedIn** [20]: A professional social network with three views, namely, college, employer and friend. Edges in the college or employer view indicate common college or workplace between users, respectively. Edges in the friend view represent LinkedIn connections. Our task deals with relationship mining to uncover latent relationship types between two nodes. Some of the node pairs have a label to indicate their relationship, such as personal community or professional contact. In total there are eight classes of relationship, and is thus a multi-class problem. Note that the relationship classes are latent and are independently labeled by the users, which do not directly correspond to existing edges in any of the three views.

**YouTube** [33]: A video-sharing network with three views, namely, friend, subscriber and video, where two users are linked if they have common friends, subscribers and favorite videos, respectively. Our task is to perform link prediction on an additional contact network, and can be formulated as a binary classification problem. Note that the friend view excludes all the contacts from the ground truth contact network. To provide negative instances, we randomly sample five times as many non-contact users for each positive contact.

**Evaluation metrics.** For binary classification on Alzheimer's and YouTube, we employ areas under the ROC and Precision-Recall (PR) curve; for multi-class classification on LinkedIn, we use micro- and macro-F scores.

**Baselines.** We compare with several common strategies (Single, Decoupled and Merged), as well as state-of-the-art methods for heterogeneous information network embedding (HIN2Vec and HeGAN) and multi-view network embedding (MVE, mvn2vec, MNE and GATNE).

**Single:** A Skip-gram model equivalent to DeepWalk applied to a single view. For each task, we run the model on every view, and report the average performance.

**Decoupled:** Each view is independently trained by the single-view Skip-gram model, with the number of dimensions set to  $\lfloor D/|V| \rfloor$ . All views' embeddings are then concatenated to form the final embedding for each node.

**Merged:** All views are merged into one network by taking the union of the edges in all views. The single-view Skip-gram model is then trained on the merged network.

**HIN2Vec** [11]: A heterogeneous network embedding approach, which samples various meta-paths and feeds them into a neural network. We merged all views to form a heterogeneous graph, where edges from different views belong to different types of relation.

**HeGAN** [15]: A heterogeneous network embedding approach, which employs the adversarial principle to learn more robust relationships between nodes. Similar to HIN2Vec, it works on a heterogeneous graph as constructed in the above.

**MVE** [28]: A multi-view network embedding approach, which promotes the first-order collaboration by regularizing the distance between view-specific embeddings and the final embedding. It also has a semi-supervised variant to learn view attention, which we denote as MVE<sup>+</sup>.

**mvn2vec** [29]: A multi-view embedding approach with two variants **mvn2vec-c** and **mvn2vec-r**. To enforce first-order collaboration, mvn2vec-c variant employs partial parameter sharing across views, whereas mvn2vec-r regularizes the distance between view-specific embeddings.

**MNE** [41]: A multi-view network embedding algorithm. For each node, the model learns a final embedding consisting of a common embedding for first-order collaboration, and view-specific relation-based embeddings.

**DMNE** [25]: A multi-view network embedding algorithm with potentially many-to-many node mappings across views, although in our setting only one-to-one mappings exist. We adopted their proximity disagreement formulation, due to its weaker assumption and better empirical performance.

**GATNE** [4]: An embedding algorithm for attributed multi-view heterogeneous network based on MNE, although in our setting we only consider one node type with multiple edge types (*i.e.*, views). Moreover, we used their transductive version since we do not consider node attributes in our setting. We compare to its variants both with and without the attention mechanism, and denote the attention-based version as GATNE<sup>+</sup>.

**Implementation details.** To sample random walks for all Skip-gram models, we applied walk length of 10, 5 walks per node, windows size of 3, and 10 samples per negative sampling. All methods adopt  $D = 128$  as the dimension of the final embedding. For the baselines, we used the implementations from their respective authors, and extensively tuned their main hyperparameters. Specifically, we chose  $\eta = 0.05$  for MVE,  $\theta = 0.8$  for mvn2vec-c and  $\gamma = 0.01$  for mvn2vec-r, and  $r = 1000$  for MNE, which give competitive results empirically. Our methods are implemented in Python using PyTorch. We set  $\alpha = \beta = 1$  for an equal weight between the three characteristics. We further set the attention hyperparameter  $\gamma = 1000$  for LinkedIn and YouTube, which is generally robust and insensitive to small changes. On the highly skewed Alzheimer's dataset, we set a smaller  $\gamma = 0.1$  to reduce reliance on the very skewed training nodes. Nonetheless, we also conducted a sensitivity study to show the impact of these hyperparameters. Finally, for all methods, the learned final embeddings are used as features to further train a logistic regression model using five-fold cross-validation. For the semi-supervised attention-based methods, only the same training folds from the downstream application is utilized during the training of node embeddings.

## 5.2 Performance Comparison

We first compare models without view attention, as most baselines do not differentiate the importance of views.

**Without view attention.** Table 4 reports the performance comparison between MANE and baselines without view attention, which treat all views uniformly. Our model MANE achieves the best performance consistently, outperforming the runner-up by up to 35.9%, 2.0% and 15.7% on the three datasets, respectively. Among the baselines, no consistent winner emerges, with mvn2vec and MNE being generally competitive.

Table 4. Performance evaluation on three datasets without view attention (bold: best; underline: runner-up).

	Alzheimer's		LinkedIn		YouTube	
Metric	ROC-AUC	PR-AUC	Micro-F	Macro-F	ROC-AUC	PR-AUC
Single	0.6968	0.0221	0.4001	0.3468	0.6334	0.1565
Decoupled	0.8200	0.0735	0.4341	0.3739	0.6700	0.1649
Merged	0.7305	0.0075	0.4197	0.3724	0.6565	<u>0.1763</u>
HIN2Vec	0.5734	0.0030	0.3014	0.2175	0.6264	0.1258
HeGAN	0.6967	0.0104	0.3937	0.3467	0.6322	0.1520
MVE	0.6543	0.0161	0.4113	0.2867	0.6276	0.1650
mvn2vec-c	0.8617	0.0890	0.4326	<u>0.3791</u>	0.6633	0.1621
mvn2vec-r	0.8756	0.0275	<u>0.4439</u>	0.3717	0.6703	0.1669
MNE	0.9195	<u>0.1676</u>	0.4334	0.3538	0.6749	0.1604
DMNE	<u>0.9357</u>	0.0603	0.3473	0.2253	0.6056	0.1394
GATNE	0.9190	0.1227	0.3202	0.1619	<u>0.6838</u>	0.1624
MANE	<b>0.9660</b>	<b>0.2277</b>	<b>0.4446</b>	<b>0.3865</b>	<b>0.6917</b>	<b>0.2039</b>

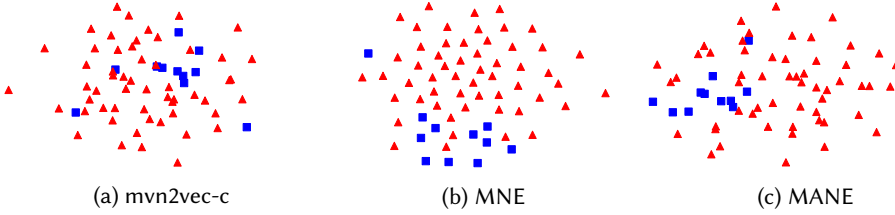


Fig. 4. Visualization of proteins: Alzheimer's positives (■) and negatives (▲). Best viewed in color.

We further visualize the embeddings generated by MANE and two competitive baselines in Fig. 4 on the Alzheimer's dataset. As the dataset is highly skewed with very few positive nodes, for a clearer visualization we uniformly sampled five times as many negative nodes as the positive nodes, while included all positive nodes. The plot of mvn2vec-c is visibly inferior, with several dispersed positive nodes far away from the others (which is substantial given that this is a skewed dataset with very few positives). MNE is improved with only one dispersed positive node, although the remaining positives are only loosely clustered. In contrast, MANE forms a relatively dense and well-defined cluster of positives.

**With view attention.** Table 5 reports the performance of the attention-based models. We make two observations. First, methods with view attention generally outperform their non-attention counterparts shown in Table 4. Second, MANE<sup>+</sup> achieves the best performance in all cases except for ROC-AUC on the highly skewed Alzheimer's dataset. However, on the same dataset MANE<sup>+</sup> outperforms the runner-up in PR-AUC by a substantial margin. Note that on highly skewed datasets, the PR curve gives a more informative assessment than the ROC curve [8].

### 5.3 Model Analysis

We further investigate various aspects of our models, including model ablation, parameter sensitivity, convergence and scalability.

**Ablation and impact of  $\alpha, \beta$ .** We examine the contribution from each order of collaboration in Eq. (7). Specifically, we evaluate (i) MANE\C1C2 without both orders of collaboration, *i.e.*,  $\alpha = \beta = 0$ ;

Table 5. Performance evaluation on three datasets with view attention (bold: best; underline: runner-up).

	Alzheimer's		LinkedIn		YouTube	
Metric	ROC-AUC	PR-AUC	Micro-F	Macro-F	ROC-AUC	PR-AUC
MVE <sup>+</sup>	0.8693	0.0067	<u>0.4480</u>	<u>0.3414</u>	0.6410	0.1544
GATNE <sup>+</sup>	<b>0.9719</b>	<u>0.1275</u>	0.3258	0.1618	<u>0.6813</u>	<u>0.1597</u>
MANE <sup>+</sup>	<u>0.9226</u>	<b>0.3380</b>	<b>0.4538</b>	<b>0.3905</b>	<b>0.7214</b>	<b>0.2422</b>

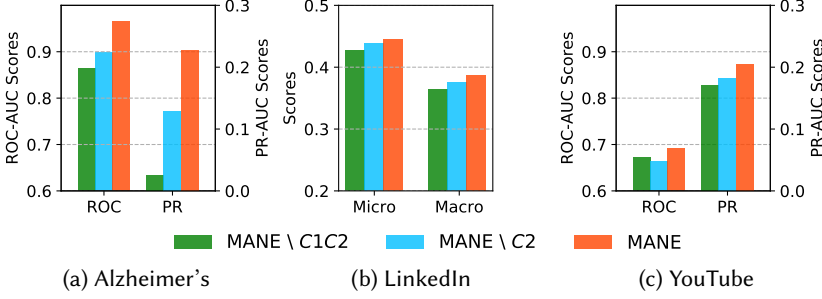
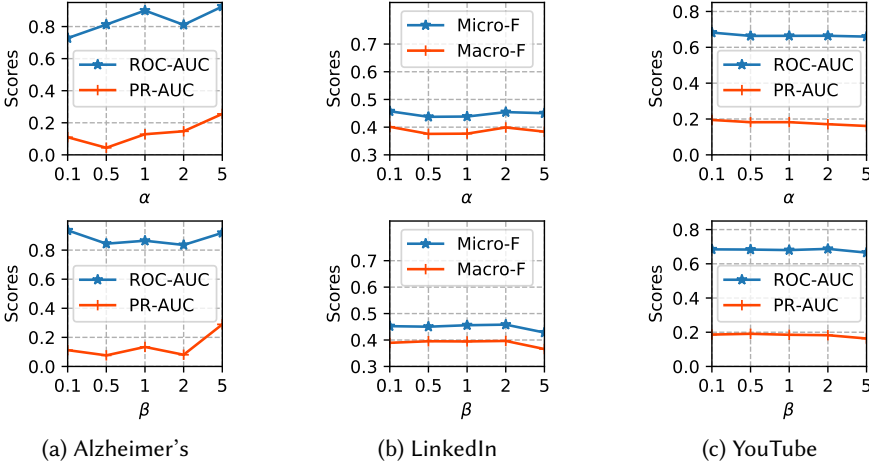


Fig. 5. Impact of first- and second-order collaboration.

Fig. 6. Impact of hyperparameters  $\alpha$  and  $\beta$ .

(ii) MANE\C2 without the second order, *i.e.*,  $\alpha = 1, \beta = 0$ . As Fig. 5 shows, MANE outperforms MANE\C2 on all three datasets, validating that the novel second-order collaboration is beneficial. In particular, the effect of the second-order collaboration is the largest on Alzheimer's, which is consistent with our analysis in Table 2. Similarly, MANE\C2 also consistently outperforms MANE\C1C2. Overall, the results justify the proposed unification of the three characteristics. For a finer-grained analysis, we further vary  $\alpha$  and  $\beta$  in Fig. 6. The performance is generally robust and stable in the range  $[0.5, 2]$  for both parameters.

**Impact of  $\gamma$ .** As discussed earlier, on highly skewed Alzheimer's we typically prefer a small  $\gamma$ , to reduce dependence on the skewed supervision. As Fig. 7(a) shows, the range  $[0.1, 10]$  exhibits

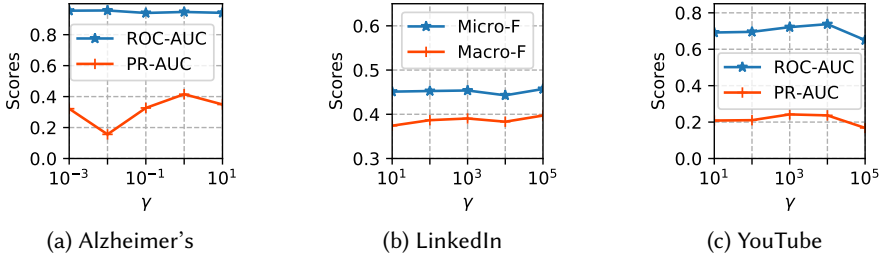
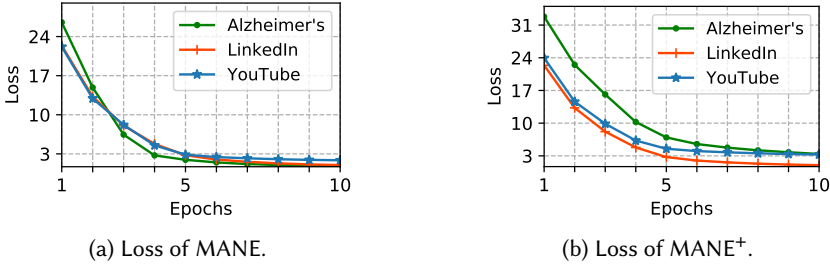
Fig. 7. Impact of hyperparameter  $\gamma$ .

Fig. 8. Convergence analysis.

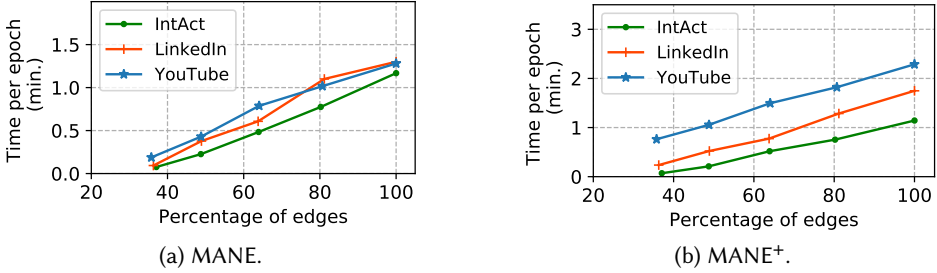


Fig. 9. Scalability analysis.

robust performance. On the other hand, on more balanced datasets YouTube and LinkedIn, a larger  $\gamma$  is often useful. As Fig. 7(b) and (c) show, the range  $[10^2, 10^4]$  gives stable outcomes.

**Convergence analysis.** Fig. 8 shows that our proposed objective functions for both MANE and MANE<sup>+</sup> converge quickly, typically between 5 and 8 epochs. MANE<sup>+</sup> often has a higher loss as it entails an additional attention-based loss component. The actual evaluation scores also demonstrate a similar convergence pattern, which are omitted for brevity. The fast convergence implies that the different loss components work well together.

**Scalability study.** We sampled differently sized multi-view networks from the original datasets. In Figure 9, we plot the time taken per epoch of training against the total number of edges in all views. We observe that both MANE and MANE<sup>+</sup> scale linearly in the total number of edges, which is consistent with our complexity analysis in Sect. 4.3. Moreover, on the IntAct and LinkedIn datasets, we observe that MANE<sup>+</sup> incurs a training duration similar to or slightly higher than MANE, as the size of training data on these two datasets are much smaller than the total number of edges. On

the other hand, on the YouTube dataset, MANE<sup>+</sup> incurs a noticeably higher training duration than MANE, due to its considerably larger training set.

## 6 CONCLUSION

In this paper, we investigated an important problem of multi-view network embedding. In addition to the traditional characteristics of diversity and (first-order) collaboration, we leveraged on the newly discovered second-order collaboration. We systematically unified the three characteristics with three categories of node pairs, and proposed MANE for multi-view collaborative network embedding. We further extended it to capture node-wise view importance using the attention mechanism in MANE<sup>+</sup>. Finally, we evaluated the performance of our proposed approaches, and demonstrated their promising results in comparison to an extensive range of state-of-the-art baselines.

## REFERENCES

- [1] Sezin Kircali Ata, Yuan Fang, Min Wu, Xiao-Li Li, and Xiaokui Xiao. 2017. Disease gene classification with metagraph representations. *Methods* 131 (2017), 83–92.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
- [3] Bokai Cao, Lifang He, Xiangnan Kong, S Yu Philip, Zhifeng Hao, and Ann B Ragin. 2014. Tensor-based multi-view feature selection with applications to brain diseases. In *ICDM*. 40–49.
- [4] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation Learning for Attributed Multiplex Heterogeneous Network. In *KDD*. 1358–1368.
- [5] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. 2015. Heterogeneous network embedding via deep architectures. In *KDD*. 119–128.
- [6] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. 2009. Multi-view clustering via canonical correlation analysis. In *ICML*. 129–136.
- [7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *JAIR* 16 (2002), 321–357.
- [8] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *ICML*. 233–240.
- [9] Paramveer Dhillion, Dean P Foster, and Lyle H Ungar. 2011. Multi-view learning of word embeddings via CCA. In *NIPS*. 199–207.
- [10] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*. 135–144.
- [11] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*. 1797–1806.
- [12] Alberto García-Durán and Mathias Niepert. 2017. Learning Graph Representations with Embedding Propagation. In *NIPS*. 5125–5136.
- [13] Derek Greene and Pádraig Cunningham. 2009. A matrix factorization approach for integrating multiple data views. In *ECML-PKDD*. 423–438.
- [14] Jingrui He and Rick Lawrence. 2011. A Graphbased Framework for Multi-Task Multi-View Learning.. In *ICML*. 25–32.
- [15] Binbin Hu, Yuan Fang, and Chuan Shi. 2019. Adversarial Learning on Heterogeneous Information Networks. In *KDD*. 120–129.
- [16] Jiaming Huang, Zhao Li, Vincent W. Zheng, Wen Wen, Yifan Yang, and Chen Yuanmi. 2018. Unsupervised Multi-view Nonlinear Graph Embedding. In *UAI*. 319–328.
- [17] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- [18] Abhishek Kumar, Piyush Rai, and Hal Daume. 2011. Co-regularized multi-view spectral clustering. In *NIPS*. 1413–1421.
- [19] Guangxia Li, Steven CH Hoi, and Kuiyu Chang. 2010. Two-view transductive support vector machines. In *SDM*. 235–244.
- [20] Rui Li, Chi Wang, and Kevin Chen-Chuan Chang. 2014. User profiling in an ego network: co-profiling attributes and relationships. In *WWW*. 819–830.
- [21] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*. 252–260.
- [22] Ye Liu, Lifang He, Bokai Cao, Philip S. Yu, Ann B. Ragin, and Alex D. Leow. 2018. Multi-View Multi-Graph Embedding for Brain Network Clustering Analysis. In *AAAI*. 117–124.



- [23] Chun-Ta Lu, Lifang He, Hao Ding, Bokai Cao, and Philip S Yu. 2018. Learning from multi-view multi-way data via structural factorization machines. In *WWW*. 1593–1602.
- [24] Guixiang Ma, Chun-Ta Lu, Lifang He, S Yu Philip, and Ann B Ragin. 2017. Multi-view graph embedding with hub detection for brain network analysis. In *ICDM*. 967–972.
- [25] Jingchao Ni, Shiyu Chang, Xiao Liu, Wei Cheng, Haifeng Chen, Dongkuan Xu, and Xiang Zhang. 2018. Co-regularized deep multi-network embedding. In *WWW*. 469–478.
- [26] Sandra Orchard, Mais Ammari, Bruno Aranda, Lionel Breuza, Leonardo Briganti, Fiona Broackes-Carter, Nancy H Campbell, Gayatri Chavali, Carol Chen, Noemi Del-Toro, et al. 2013. The MIntAct project—IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research* 42, D1 (2013), D358–D363.
- [27] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online learning of social representations. In *KDD*. 701–710.
- [28] 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*. 1767–1776.
- [29] 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).
- [30] Yiwei Sun, Ngot Bu, Tsung-Yu Hsieh, and Vasant Honavar. 2018. Multi-view network embedding via graph factorization clustering and co-regularized multi-view agreement. In *ICDM Workshop on Graph Analytics*. 1006–1013.
- [31] Yizhou Sun and Jiawei Han. 2013. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explor.* 14, 2 (2013), 20–28.
- [32] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW*. 1067–1077.
- [33] Lei Tang, Xufei Wang, and Huan Liu. 2009. Uncovering groups via heterogeneous interaction analysis. In *ICDM*. 503–512.
- [34] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
- [35] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. GraphGan: Graph representation learning with generative adversarial nets. In *AAAI*. 2508–2515.
- [36] James Z Wang, Zhidian Du, Rapeeporn Payattakool, Philip S Yu, and Chin-Fu Chen. 2007. A new method to measure the semantic similarity of GO terms. *Bioinformatics* 23, 10 (2007), 1274–1281.
- [37] Yueyang Wang, Liang Hu, Yueting Zhuang, and Fei Wu. 2018. Intra-view and inter-view attention for multi-view network embedding. In *PCM*. 201–211.
- [38] Jia Wu, Shirui Pan, Xingquan Zhu, Chengqi Zhang, and S Yu Philip. 2018. Multiple structure-view learning for graph classification. *TNNLS* 29, 7 (2018), 3236–3251.
- [39] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634* (2013).
- [40] Dejian Yang, Senzhang Wang, Chaozhuo Li, Xiaoming Zhang, and Zhoujun Li. 2017. From properties to links: Deep network embedding on incomplete graphs. In *CIKM*. 367–376.
- [41] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. 2018. Scalable Multiplex Network Embedding. In *IJCAI*. 3082–3088.
- [42] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view knowledge graph embedding for entity alignment. In *IJCAI*. 5429–5435.
- [43] Wentao Zhang, Yuan Fang, Zemin Liu, Min Wu, and Xinming Zhang. 2020. mg2vec: Learning Relationship-Preserving Heterogeneous Graph Representations via Metagraph Embedding. *TKDE* (2020).
- [44] Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. 2010. Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. *Physica A: Stat. Mech. and Appl.* 389, 1 (2010), 179–186.
- [45] Vincent W. Zheng, Mo Sha, Yuchen Li, Hongxia Yang, Yuan Fang, Zhenjie Zhang, Kian-Lee Tan, and Kevin Chen-Chuan Chang. 2018. Heterogeneous Embedding Propagation for Large-Scale E-Commerce User Alignment. In *ICDM*. 1434–1439.