mvn2vec: Preservation and Collaboration in Multi-View Network Embedding

Yu Shi[†], Fangqiu Han[‡], Xinwei He[†], Xinran He[‡], Carl Yang[†], Luo Jie[‡], Jiawei Han[†]
[†]University of Illinois at Urbana-Champaign, IL, USA [‡]Snap Inc., Los Angeles, CA, USA
[†]{yushi2, xhe17, jiyang3, hanj}@illinois.edu [‡]{fangqiu.han, xhe2, roger.luo}@snap.com

ABSTRACT

Multi-view networks are broadly present in real-world applications. In the meantime, network embedding has emerged as an effective representation learning approach for networked data. Therefore, we are motivated to study the problem of multi-view network embedding with a focus on the optimization objectives that are specific and important in embedding this type of networks. In our practice of embedding real-world multi-view networks, we explicitly identify two such objectives, which we refer to as *preservation* and *collaboration*. The in-depth analysis of these two objectives are discussed throughout this paper. In addition, the novel MVN2VEC algorithms are proposed to (i) study how varied extent of *preservation* and *collaboration* can impact embedding learning and (ii) explore the feasibility of achieving better embedding quality by modeling them simultaneously. With experiments on a large-scale internal Snapchat dataset, two public datasets and a series of synthetic datasets, we confirm the validity and importance of *preservation* and *collaboration* as two objectives for multi-view network embedding. These experiments further demonstrate that better embedding can be obtained by simultaneously modeling the two objectives, while not over-complicating the model or requiring additional supervision. The code and the processed datasets are available at https://yu-shi-homepage.github.io/.

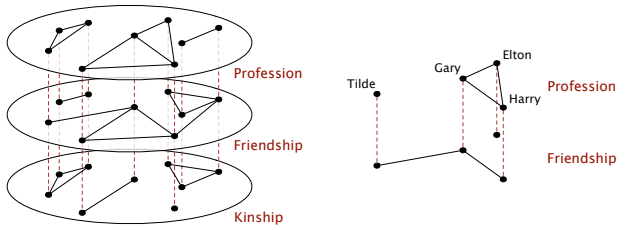
KEYWORDS

Multi-view networks, network embedding, graph mining, representation learning

ACM Reference Format:
Yu Shi[†], Fangqiu Han[‡], Xinwei He[†], Xinran He[‡], Carl Yang[†], Luo Jie[‡], Jiawei Han[†]. 2022. mvn2vec: Preservation and Collaboration in Multi-View Network Embedding. In *Proceedings of ACM Conference (MLG'22).* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In real-world applications, objects can be associated with different types of relations. These objects and their relationships can be naturally represented by multi-view networks, *i.e.*, multiplex networks or multi-view graphs [4, 9, 17, 18, 23, 32]. Figure 1a gives a toy multi-view network, where each view corresponds to a type of edge, and all views share the same set of nodes. As a more concrete example, a four-view network of users can be used to describe a social networking service with social relationship and interaction including friendship, following, message exchange, and post viewing. With the vast availability of multi-view networks, it is of interest to mine such networks.



(a) A toy multi-view network. **(b) A close-up look of profession view and friendship view.**

Figure 1: A toy example of multi-view networks where each node represents a person and the three views correspond to three types of interpersonal relations. Co-workers are linked in the profession view, friends are linked in the friendship view, and relatives are linked in the kinship view.

In the meantime, network embedding has emerged as a scalable representation learning method for networked data [7, 19, 24, 25]. Specifically, network embedding projects nodes of networks into the embedding spaces. With the semantic information of each node encoded, the learned embedding can be directly used as features in various downstream applications [7, 19, 24]. Motivated by the success of network embedding for homogeneous networks [7, 16, 19, 20, 24, 25], where nodes and edges are untyped, we believe it is important to better understand multi-view network embedding.

To design embedding method for multi-view networks, the primary challenge lies in how to use the type information on edges from different views. As a result, we are interested in investigating the following two questions:

(1) With the availability of multiple edge types, what are the objectives that are *specific* and *important* to multi-view network embedding?
(2) Can we achieve better embedding quality by modeling these objectives jointly?

To answer the first question, we identify two such objectives, *preservation* and *collaboration*, from our practice of embedding real-world multi-view networks. **Collaboration** – In some datasets, edges between the same pair of nodes may be observed in different views due to shared latent reasons. For instance, in a social network,

MLG'22, August 15 2022, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

if we observe an edge between a user pair in either the message exchange view or the post viewing view, likely these two users are happy to be associated with each other. In such scenario, these views may complement each other, and embedding them jointly may potentially yield better results than embedding them independently. We call such synergetic effect in jointly embedding multiple views by *collaboration*, which is also the primary intuition behind most existing multi-view network algorithms [9, 18, 23, 32]. **Preservation** – On the other hand, it is possible for different network views to have different semantic meanings; it is also possible that a portion of nodes has completely disagreeing edges in different views since edges in different views are formed due to unrelated latent reasons. For example, the professional relationship may not always align well with friendship. If we embed the profession view and the friendship view in Figure 1b into the same embedding space, the embedding of Gary will be close to both Tilde and Elton. As a result, the embedding of Tilde will also not be too distant from Elton due to transitivity. However, this is not a desirable result, because Tilde and Elton are not closely related regarding either profession or friendship according to the original multi-view network. In other words, embedding in this way fails to preserve the unique information carried by different network views. We refer to such need for preserving unique information carried by different views as *preservation*. The detailed discussion of the validity and importance of *preservation* and *collaboration* is presented in Section 4.

Furthermore, the need for *preservation* and *collaboration* may co-exist in the same multi-view network. Two scenarios can result in this situation: (i) a pair of views are generated from very similar latent reason, while another pair of views carries completely different semantic meanings; and more subtly (ii) for the same pair of views, one portion of nodes has consistent edges in different views, while another portion of nodes have totally disagreeing edges in different views. One example of the latter scenario is that professional relationship does not align well with friendship in some cultures, whereas co-workers often become friends in certain other cultures [1]. Therefore, we are also interested in exploring the feasibility of achieving better embedding quality by modeling *preservation* and *collaboration* simultaneously, and we address this problem in Section 5 and beyond.

We note that, instead of proposing a sophisticated model that beats many baselines, this paper focus on the objectives of interests for multi-view network embedding. In experiments, we compare methods with a highlight on the roles the two objectives play in different scenarios. For the same reason, the scenarios where additional supervision is available are excluded from this paper, while the lessons learned from the unsupervised scenario can also be applied to the supervised multi-view network embedding algorithms. Additionally, node embedding learned by an unsupervised approach can directly apply to different downstream tasks, while supervised algorithms yield embedding specifically good for tasks where the supervision comes from. We summarize our contributions as follows.

(1) We study the objectives that are specific and important to multi-view network embedding and identify *preservation* and *collaboration* as two such objectives from the practice of embedding real-world multi-view networks.
(2) We explore the feasibility of attaining better embedding by simultaneously modeling *preservation* and *collaboration*, and propose two multi-view network embedding methods – MVN2VEC-CON and MVN2VEC-REG.
(3) We conduct experiments with various downstream applications on three datasets. These experiments corroborate the validity and importance of *preservation* and *collaboration* and demonstrate the effectiveness of the proposed methods.

2 RELATED WORK

An attention-based collaboration framework is proposed for multi-view network embedding [21]. The problem setting of this work differs from ours since it requires supervision for its attention mechanism. Besides, this approach does not directly model *preservation* – one of the objectives that we deem important for multi-view network embedding. since the final embedding derived via linear combination in this framework is a trade off between representations from all views. A deep learning architecture has also been proposed for embedding multi-networks [14], where the multi-network is a more general concept than the multi-view network and allows many-to-many correspondence across networks. While the proposed model can be applied to the more specific multi-view networks, it does not focus on the study of the objectives of multi-view network embedding. Another group of related work studies the problem of jointly modeling multiple network views using latent space models [5, 6]. These works again do not model *preservation*. Two recent papers are proposed based on our method. Xiong et al. [27] propose a contrastive learning based embedding method using tensorized attention to adaptively capture multiplex information. Ata en al. [2] propose a multi-view embedding method that considers second order *collaboration* on top of the first order *collaboration* and *preservation*. However these works do not focus on detailed study of *preservation* and *collaboration*. There exist a few more studies that touches the topic of multi-view network embedding[3, 11, 12, 15, 28, 30] They do not model *preservation* and *collaboration* or do not aim to provide in-depth study of these objectives. Most recently, a new line of works emerged using multi-view network with self-supervised contrastive learning for network embedding [8, 26, 31]. Zhao et al. [31] propose a network embedding method by sampling and encoding subgraphs of different views using graph neural networks and capturing the intra-view and inter-view information by multi-view contrastive learning. However, these works do not aim to provide in-depth study of *preservation* and *collaboration*.

3 PRELIMINARIES

Definition 3.1 (Multi-View Network). *A multi-view network $G = (\mathcal{U}, \{\mathcal{E}^{(v)}\}_{v\in\mathcal{V}})$ is a network consisting of a set $\mathcal{U}$ of nodes and a set $\mathcal{V}$ of views, where $\mathcal{E}^{(v)}$ consists of all edges in view $v \in \mathcal{V}$. If a multi-view network is weighted, then there exists a weight mapping $w : \{\mathcal{E}^{(v)}\}_{v\in\mathcal{V}} \to \mathbb{R}$ such that $w^{(v)}_{uu'} := w(e^{(v)}_{uu'})$ is the weight of the edge $e^{(v)}_{uu'} \in \mathcal{E}^{(v)}$, which joints nodes $u \in \mathcal{U}$ and $u' \in \mathcal{U}$ in view $v \in \mathcal{V}$.*

Additionally, when context is clear, we use the network view v of $G = (\mathcal{U}, \{\mathcal{E}^{(v)}\}_{v\in\mathcal{V}})$ to denote the network $G^{(v)} = (\mathcal{U}, \mathcal{E}^{(v)})$.

Table 1: Summary of symbols

Symbol	Definition
$\mathcal{V}$	The set of all network views
$\mathcal{U}$	The set of all nodes
$\mathcal{E}^{(v)}$	The set of all edges in view $v \in \mathcal{V}$
$\mathcal{W}^{(v)}$	The list of random walk pairs from view $v \in \mathcal{V}$
$\mathbf{f}_u$	The final embedding of node $u \in \mathcal{U}$
$\mathbf{f}_u^v$	The center embedding of node $u \in \mathcal{U}$ *w.r.t.* view $v \in \mathcal{V}$
$\tilde{\mathbf{f}}_u^v$	The context embedding of node $u \in \mathcal{U}$ *w.r.t.* view $v \in \mathcal{V}$
$\theta \in [0, 1]$	The hyperparameter on parameter sharing in MVN2VEC-CON
$\gamma \in \mathbb{R}_{\geq 0}$	The hyperparameter on regularization in MVN2VEC-REG
$D \in \mathbb{N}$	The dimension of the embedding space

Definition 3.2 (Network Embedding). *Network embedding aims at learning a (center) embedding $\mathbf{f}_u \in \mathbb{R}^D$ for each node $u \in \mathcal{U}$ in a network, where $D \in \mathbb{N}$ is the dimension of the embedding space.*

Besides the center embedding $\mathbf{f}_u \in \mathbb{R}^D$, a family of popular algorithms [13, 24] also deploy a context embedding $\tilde{\mathbf{f}}_u \in \mathbb{R}^D$ for each node u. Moreover, when the learned embedding is used as the feature vector for downstream applications, we take the center embedding of each node as feature following the common practice in algorithms involving context embedding.

4 PRESERVATION AND COLLABORATION IN MULTI-VIEW NETWORK EMBEDDING

In this section, we elaborate on the intuition and presence of *preservation* and *collaboration* – the two objectives introduced in Section 1. We first describe and investigate the motivating phenomena observed in our practice of embedding real-world multi-view networks, and then discuss how they can be explained by the two proposed objectives.

Two straightforward approaches for embedding multi-view networks. To extend untyped network embedding algorithms to multi-view networks, two straightforward yet practical approaches exist. We refer to these two approaches as the *independent* model and the *one-space* model. Specifically, we denote $\mathbf{f}_u^v \in \mathbb{R}^{D_v}$ the embedding of node $u \in \mathcal{U}$ achieved by embedding only the view $v \in \mathcal{V}$ of the multi-view network, where D_v is the dimension of the embedding space for network view v. With such notation, the *independent* model and the *one-space* model are briefly introduced as follows, while further details can be found in Section 6.2.

- **The independent model**. Embed each view independently, and then concatenate to derive the final embedding $\mathbf{f}_u$:

$$\mathbf{f}_u = \bigoplus_{v \in \mathcal{V}} \mathbf{f}_u^v \in \mathbb{R}^D, \tag{1}$$

  where $D = \sum_{v \in \mathcal{V}} D_v$, and $\bigoplus$ represents concatenation. In other words, the embedding of each node in the *independent* model resides in the direct sum of multiple embedding spaces. *This approach preserves the information embodied in each view, but do not allow collaboration across different views in the embedding learning process.*
- **The one-space model**. Let embedding for different views share parameters when learning the final embedding $\mathbf{f}_u$:

$$\mathbf{f}_u = \mathbf{f}_u^v \in \mathbb{R}^D, \ \forall v \in \mathcal{V}. \tag{2}$$

Table 2: Embedding quality of two real-world multi-view networks using the *independent* model and the *one-space* model.

Dataset	Metric	*independent*	*one-space*
YouTube	ROC-AUC	**0.931**	0.914
	PRC-AUC	**0.745**	0.702
Twitter	ROC-AUC	0.724	**0.737**
	PRC-AUC	0.447	**0.466**

  Therefore, the final embedding space correlates with all network views. *This approach allows different views to collaborate in learning a unified embedding, but do not preserve information specifically carried by each view.* This property of the *one-space* model is corroborated by additional experiment in Section F.

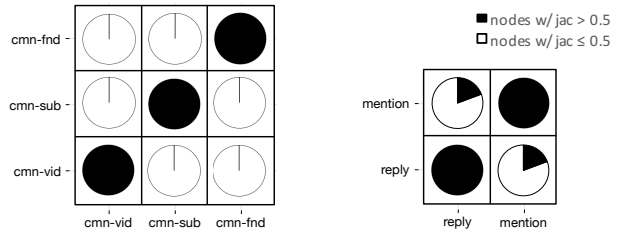
Further details on these models can be found in Section 6.2. It should also be noted that the embedding learned by the *one-space* model cannot be obtained by linearly combining $\{\mathbf{f}_u^v\}_{v \in \mathcal{V}}$ in the *independent* model. This is because most network embedding models are non-linear.

Embedding real-world multi-view networks by straightforward approaches. Two networks, YouTube and Twitter, are used in these exploratory experiments with users being nodes on each network. YouTube has three views representing common videos (cmn-vid), common subscribers (cmn-sub), and common friends (cmn-fnd) shared by each user pair, while Twitter has two views corresponding to replying (reply) and mentioning (mention) among users. The downstream evaluation task is to infer whether two users are friends, and the results are presented in Table 2.

It can be seen that neither straightforward approach is categorically better than the other. In particular, the *independent* model consistently outperformed the *one-space* model in the YouTube experiment, while the *one-space* model outperformed the *independent* model in Twitter. Furthermore, we interpret the varying performance of the two approaches by the varying extent of needs for modeling *preservation* and modeling *collaboration* when embedding different networks. Specifically, recall that the *independent* model only captures *preservation*, while *one-space* only captures *collaboration*. As a result, we speculate if a certain dataset craves for more *preservation* than *collaboration*, the *independent* model will outperform the *one-space* model. Otherwise, the *one-space* model will win.

To corroborate our interpretation of the results, we examine the involved datasets and look into the agreement between information carried by different network views. We achieve this by a Jaccard coefficient based measurement, where the Jaccard coefficient is a similarity measure with range $[0, 1]$, defined as $J(\mathcal{S}_1, \mathcal{S}_2) = |\mathcal{S}_1 \cap \mathcal{S}_2| / |\mathcal{S}_1 \cup \mathcal{S}_2|$ for set $\mathcal{S}_1$ and set $\mathcal{S}_2$. For a pair of network views, a node can be connected to a different set of neighbors in each of the two views. We calculate the Jaccard coefficient between these two sets of neighbors. In Figure 2, we apply this measurement on both datasets and illustrate the proportion of nodes with the Jaccard coefficient greater than 0.5 for each view pair.

As presented in Figure 2, little agreement exists between each pair of different views on YouTube. As a result, it is not surprising that *collaboration* among different views is not as needed as



(a) YouTube. **(b) Twitter.**

Figure 2: Agreement between information carried by each pair of network views given by a Jaccard coefficient based measurement.

preservation. On the other hand, a substantial portion of nodes has Jaccard coefficient greater than 0.5 over different views on Twitter – not surprising to see modeling *collaboration* brings about more benefits than modeling *preservation*.

5 THE MVN2VEC MODELS

In the previous section, *preservation* and *collaboration* are identified as important objectives for multi-view network embedding. In the extreme cases, where only *preservation* is needed – each view carries a distinct semantic meaning – or only *collaboration* is needed – all views carry the same semantic meaning – simply choosing between *independent* and *one-space* may be enough to generate satisfactory results. However, it is of more interest to study the scenario where both *preservation* and *collaboration* co-exist in a multi-view network. Therefore, we are motivated to (i) study how varied extent of *preservation* and *collaboration* can impact embedding learning and (ii) explore the feasibility of achieving better embedding by simultaneously modeling both objectives. To this end, we propose two methods that capture both objectives, while not over-complicating the model or requiring additional supervision. These two approaches are named MVN2VEC-CON and MVN2VEC-REG, where MVN2VEC is short for **m**ulti-**v**iew **n**etwork **to vec**tor, while CON and REG stand for constrained and regularized.

As with the notation convention in Section 4, we denote $\mathbf{f}_u^v \in \mathbb{R}^{D_v}$ and $\tilde{\mathbf{f}}_u^v \in \mathbb{R}^{D_v}$ the center and context embedding, respectively, of node $u \in \mathcal{U}$ for view $v \in \mathcal{V}$. Further given the network view v, *i.e.*, $G^{(v)} = (\mathcal{U}, \mathcal{E}^{(v)})$, we use an intra-view loss function to measure how well the current embedding can represent the original network view

$$fl(\{\mathbf{f}_u^v, \tilde{\mathbf{f}}_u^v\}_{u\in\mathcal{U}} | G^{(v)}). \tag{3}$$

We defer the detailed definition of this loss function to a later point of this section. We let $D_v = D/|\mathcal{V}| \in \mathbb{N}$ for all $v \in \mathcal{V}$ out of convenience for model design. To further incorporate multiple views with the intention to model both *preservation* and *collaboration*, two approaches are proposed as follows.

MVN2VEC-CON. The MVN2VEC-CON model does not enforce further design on the center embedding $\{\mathbf{f}_u^v\}_{u\in\mathcal{U}}$ in the hope of preserving the semantics of each individual view. To reflect *collaboration*, MVN2VEC-CON enforce constraints on the context embedding $\{\tilde{\mathbf{f}}_u^v\}_{u\in\mathcal{U}}$ for parameter sharing across different views, so they are required to have the form

$$\tilde{\mathbf{f}}_u^v = \varphi_\theta^v(\{\tilde{\mathbf{g}}_u^{v'}\}_{v'\in\mathcal{V}}) \coloneqq (1-\theta)\cdot\tilde{\mathbf{g}}_u^v + \frac{\theta}{|\mathcal{V}|}\cdot\sum_{v'\in\mathcal{V}}\tilde{\mathbf{g}}_u^{v'}, \tag{4}$$

where $\tilde{\mathbf{g}}_u^v \in \mathbb{R}^{D_v}$ and $\theta \in [0,1]$ is a hyperparameter controlling the extend to which model parameters are shared. The greater the value of θ, the more the model enforces parameter sharing and thereby encouraging more *collaboration* across different views. The MVN2VEC-CON model solves the following optimization problem

$$\min_{\{\mathbf{f}_u^v, \tilde{\mathbf{g}}_u^v\}} \sum_{v\in\mathcal{V}} l(\{\mathbf{f}_u^v, \varphi_\theta^v(\{\tilde{\mathbf{g}}_u^{v'}\}_{v'\in\mathcal{V}})\}_{u\in\mathcal{U}} | G^{(v)}). \tag{5}$$

After model learning, the final embedding for node u is given by $\mathbf{f}_u = \bigoplus_{v\in\mathcal{V}} \mathbf{f}_u^v$. We note that in the extreme case when θ is set to be 0, the model will be identical to the *independent* model in Section 4. To further distinguish the importance of different views, one can replace θ in Eq. (4) with a view-specific parameter θ_v, we defer the study of which to future work.

MVN2VEC-REG. Instead of setting hard constraints on how parameters are shared across different views, the MVN2VEC-REG model regularizes the embedding across different views and solves the following optimization problem

$$\min_{\{\mathbf{f}_u^v, \tilde{\mathbf{f}}_u^v\}} \sum_{v\in\mathcal{V}} l(\{\mathbf{f}_u^v, \tilde{\mathbf{f}}_u^v\}_{u\in\mathcal{U}} | G^{(v)}) + \gamma\cdot[\mathcal{R}^v + \tilde{\mathcal{R}}^v], \tag{6}$$

where $\gamma \in \mathbb{R}_{\geq 0}$ is a hyperparameter, $\mathcal{R}^v = \sum_{u\in\mathcal{U}} \|\mathbf{f}_u^v - \frac{1}{|\mathcal{V}|}\sum_{v'\in\mathcal{V}} \mathbf{f}_u^{v'}\|_2^2$, $\tilde{\mathcal{R}}^v = \sum_{u\in\mathcal{U}} \|\tilde{\mathbf{f}}_u^v - \frac{1}{|\mathcal{V}|}\sum_{v'\in\mathcal{V}} \tilde{\mathbf{f}}_u^{v'}\|_2^2$, and $\|\cdot\|_2$ is the l-2 norm. This model captures *preservation* again by letting $\{\mathbf{f}_u^v\}_{u\in\mathcal{U}}$ and $\{\tilde{\mathbf{f}}_u^v\}_{u\in\mathcal{U}}$ to reside in the embedding subspace specific to view $v \in \mathcal{V}$, while each of these subspaces are distorted via cross-view regularization to model *collaboration*. Similar to the MVN2VEC-CON model, the greater the value of the hyperparameter γ, the more the *collaboration* is encouraged, and the model is identical to the *independent* model when $\gamma = 0$. To distinguish the importance of different views, one can replace $(\sum_{v'\in\mathcal{V}} \mathbf{f}_u^{v'})/|\mathcal{V}|$ in the formulation of $\mathcal{R}^v$ with $(\sum_{v'\in\mathcal{V}} \lambda_{v'}\cdot\mathbf{f}_u^{v'})/(\sum_{v'\in\mathcal{V}} \lambda_{v'}|)$, and revise $\tilde{\mathcal{R}}^v$ similarly. We also defer this study to future work for simplicity.

Intra-view loss function. There are many possible approaches to formulate the intra-view loss function in Eq. (3). In our framework, we adopt the random walk plus skip-gram approach, which is one of the most common in the literature [7, 19, 20]. Specifically, for each view $v \in \mathcal{V}$, multiple rounds of random walks are sampled starting from each node in $G^{(v)} = (\mathcal{U}, \mathcal{E}^{(v)})$. In any random walk, a node $u \in \mathcal{U}$ and a neighboring node $n \in \mathcal{U}$ constitute one random walk pair, and a list $\mathcal{W}^{(v)}$ of random walk pairs can thereby be derived. We will describe the detailed description of the generation of $\mathcal{W}^{(v)}$ later in this section. The intra-view function is then given by

$$l(\{\mathbf{f}_u^v, \tilde{\mathbf{f}}_u^v\}_{u\in\mathcal{U}} | G^{(v)}) = -\sum_{(u,n)\in\mathcal{W}^{(v)}} \log p^{(v)}(n|u), \tag{7}$$

where $p^{(v)}(n|u) = \exp\left(\mathbf{f}_u^v\cdot\tilde{\mathbf{f}}_n^v\right)/\sum_{n'\in\mathcal{U}}\exp\left(\mathbf{f}_u^v\cdot\tilde{\mathbf{f}}_{n'}^v\right)$.

Model inference. To optimize the objectives in Eq. (5) and (6), we opt to asynchronous stochastic gradient descent (ASGD) [22] following existing skip-gram–based algorithms [7, 13, 19, 20, 24].

In this regard, $\mathcal{W}^{(v)}$ from all views are joined and shuffled to form a new list $\mathcal{W}$ of random walk pairs for all views. Then each step of ASGD draws one random walk pair from $\mathcal{W}$ and updates corresponding model parameters with one-step gradient descent. Negative sampling is adopted as in other skip-gram–based methods [7, 13, 19, 20, 24], which approximates $\log p^{(v)}(n|u)$ in Eq. (7) by $-\log\sigma(\mathbf{f}_u^v \cdot \tilde{\mathbf{f}}_n^v) - \sum_{i=1}^{K} \mathbb{E}_{n_i' \sim P^{(v)}} \log\sigma(-\mathbf{f}_u^v \cdot \tilde{\mathbf{f}}_{n_i'}^v)$, where $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function, K is the negative sampling rate, $P^{(v)}(u) \propto \left[D_u^{(v)}\right]^{3/4}$ is the noise distribution, and $D_u^{(v)}$ is the number of occurrences of node u in $\mathcal{W}^{(v)}$ [13].

With negative sampling, the objective function involving one walk pair (u, n) from view v in MVN2VEC-CON is

$$O_{\text{CON}} = \log\sigma(\mathbf{f}_u^v \cdot \varphi_\theta^v(\{\tilde{\mathbf{g}}_n^{v'}\}_{v' \in \mathcal{V}})) + \sum_{i=1}^{K} \mathbb{E}_{n_i' \sim P^{(v)}} \log\sigma(-\mathbf{f}_u^v \cdot \varphi_\theta^v(\{\tilde{\mathbf{g}}_{n_i'}^{v'}\}_{v' \in \mathcal{V}})).$$

On the other hand, the objective function involving (u, n) from view v in MVN2VEC-REG is

$$O_{\text{REG}} = \log\sigma(\mathbf{f}_u^v \cdot \tilde{\mathbf{f}}_n^v) + \gamma \cdot (\mathcal{R}_u^v + \tilde{\mathcal{R}}_n^v) + \sum_{i=1}^{K} \left[\mathbb{E}_{n_i' \sim P^{(v)}} \log\sigma(-\mathbf{f}_u^v \cdot \tilde{\mathbf{f}}_{n_i'}^v) + \gamma \cdot (\mathcal{R}_u^v + \tilde{\mathcal{R}}_{n_i'}^v)\right].$$

The gradients of the above two objective function used for ASGD are provided in Appendix A.

Lastly, we also describe the details on **random walk pair generation** in Appendix B, provide **complexity analysis** in Appendix C, and summarize the MVN2VEC algorithms in Appendix D.

6 EXPERIMENTS

In this section, we further corroborate the intuition of *preservation* and *collaboration*, and demonstrate the feasibility of simultaneously model these two objectives.

We introduce the real-world datasets, baselines, and experiment setting for comprehensive quantitative evaluations. Then, we analyze the evaluation results and provide further discussion. Additionally, to validate our intuition discussed above, we perform a **case study** on a series of synthetic multi-view networks that have varied extent of *preservation* and *collaboration* in Appendix E.

6.1 Data Description and Evaluation Tasks

We perform quantitative evaluations on three real-world multi-view networks: Snapchat, YouTube, and Twitter. The key statistics are summarized in Table 3, and we describe these datasets as follows.

YouTube. YouTube is a video-sharing website. We use the YouTube dataset made publicly available by the Social Computing Data Repository [29][1]. From this dataset, a network with three views is constructed, where each node is a core user, and the edges in the three views represent the number of common friends, the number of common subscribers, and the number of common favorite videos, respectively. Note that the core users are those from which the author of the dataset crawled the data, and their friends can fall out of the scope of the set of core users. Without user label available for classification, we perform only link prediction task on top of the user embedding. This task aims at inferring whether two core users are friends, which has also been used for evaluation by existing research [21]. Each core user forms positive pairs with his or her core friends, and we randomly select 5 times as many non-friend core users to form negative examples. Records are split into training, validation, and test sets as in the link prediction task on YouTube.

Twitter. Twitter is an online news and social networking service. We use the Twitter dataset made publicly available by the Social Computing Data Repository [10][2]. From this dataset, a network with two views is constructed, where each node is a user, and the edges in the two views represent the number of replies and the number of mentions, respectively. Again, we evaluate using a link prediction task that infers whether two users are friends as in existing research [21]. Negative examples generation and training–validation–test split method are the same as in the YouTube dataset.

Snapchat. Snapchat is a multimedia social networking service. On the Snapchat multi-view social network, each node is a user, and the three views correspond to friendship, chatting, and story viewing[3]. We perform experiments on the sub-network consisting of all users from Los Angeles. The data used to construct the network are collected from two consecutive weeks in the Spring of 2017. Additional data for downstream evaluation tasks are collected from the following week (week 3). We perform a multi-label classification task, and a link prediction task on top of the same user embedding learned from each network. For classification, we classify whether or not a user views each of the 10 most popular discover channels[4] according to the user viewing history in week 3, which aims at inferring users' preference and thereby guide product design in content serving. For each channel, the users who view this channel are labeled positive, and we randomly select 5 times as many users who do not view this channel as negative examples. These records are then randomly split into training, validation, and test sets. This is a multi-label classification problem that aims at inferring users' preference on different discover channels and can therefore guide product design in content serving. For link prediction, we predict whether two users would view the stories posted by each other in week 3, which aims to estimate the likelihood of story viewing between friends, so as to re-rank stories accordingly. Negative examples are the users who are friends but do not have story viewing in the same week. It is worth noting that this definition yields more positive examples than negative examples, which is the cause of a relatively high AUPRC score observed in experiments. These records are then randomly split into training, validation, and test sets with the constraint that a user appears as the viewer of a record in at most one of the three sets. This task aims to estimate the likelihood of story viewing between friends so that the application can rank stories accordingly.

We also provide the Jaccard coefficient based measurement on Snapchat in Figure 3. It can be seen that the cross-view agreement between each pair of views in the Snapchat network falls in between YouTube and Twitter as presented previously in Section 2.

[1] http://socialcomputing.asu.edu/datasets/YouTube

[2] https://snap.stanford.edu/data/higgs-twitter.html

[3] https://support.snapchat.com/en-US/a/view-stories

[4] https://support.snapchat.com/en-US/a/discover-how-to

Table 3: Basic statistics for the three real-world networks, where the number of edges specifies the total number of edges from all network views.

Dataset	# views	# nodes	# edges
Snapchat	3	7,406,859	131,729,903
YouTube	3	14,900	7,977,881
Twitter	2	116,408	183,341

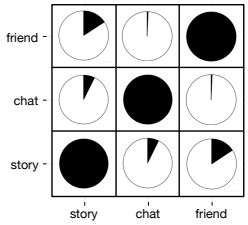


Figure 3: Agreement between each pair of views for Snapchat.

For each evaluation task on all three networks, training, validation, and test sets are derived in a shuffle split manner with an 80%–10%–10% ratio. We split the data and repeat the experiments 20 times to compute the mean and its standard error under each metric. Furthermore, a node is excluded from evaluation if it is isolated from other nodes in at least one of the multiple views. This processing ensures every node has a valid representation when embedding only one network view.

6.2 Baselines and Experimental Setup

In this section, we describe the baselines as well as experimental setup for both embedding learning and downstream evaluation tasks.

Baselines. Quantitative evaluation results are obtained by applying downstream learner upon embedding learned by a given embedding method. Therefore, for fair comparisons, we use the same downstream learner in the same evaluation task. We experiment with both in-house baselines and external baselines from related work. Since our study aims at understanding the objectives of multi-view network embedding, we build the following in-house embedding methods from the same random work plus skip-gram approach with the same model inference method as described in Section 5. **Independent** – As introduced in Section 4, the *independent* model is equivalent to MVN2VEC-CON when $\theta = 0$, and to MVN2VEC-REG when $\gamma = 0$. It preserves the information embodied in each view, but do not allow collaboration across different views in the embedding process. **One-space** – Also introduced in Section 4, the *one-space* enables different views to collaborate in learning a unified embedding, but do not preserve information specifically carried by each view. **View-merging** – The *view-merging* model first merges all network views into one unified view, and then learn the embedding of this single unified view. To comply with the assumed equal importance of all network views, we rescale the weights of edges proportionally in each view to the same total weight. This method serves as an alternate approach to *one-space* in modeling *collaboration*. The difference between *view-merging* and *one-space* essentially lies in whether or not random walks can cross different views. We note that just like *one-space*, *view-merging* does not model *preservation*. **Single-view** – For each network view, the *single-view* model learns embedding from only one view as a sanity check to verify whether introducing more than one view does bring in informative signals in each evaluation task. In other words, it is identical to embedding only one of the multiple network views using the DeepWalk algorithm [19], or equivalently node2vec with both return parameter and in-out parameter set to 0 [7]. This baseline is used to as a sanity check to verify whether introducing more than one view does bring in informative signals in each evaluation task.

We also experiment with two external baselines that have executable implementation released by the original authors and can be applied to our experiment setting with minimal tweaks. Note that the primary observations and conclusions made in the experiments would still hold without comparing to these external baselines. **MVE** [21] – An attention-based semi-supervised method for multi-view network embedding. The released implementation supports supervision from classification tasks. For fair comparison, in our unsupervised link prediction experiments, we assign the same dummy class label to 1% randomly selected nodes. We have observed that the percentage of nodes receiving such dummy supervision does not significantly affect the evaluation results in additional experiments. **DMNE** [14] – A deep learning architecture for multi-network embedding, where the multi-network is a more general concept than the multi-view network. As a more complex model, the released DMNE implementation takes significantly more time to train on our datasets, and we hence use only the default hyperparameters. The authors commented in the original paper that this implementation can be further sped up with additional parallelization. We experiment with two external baselines on the smaller YouTube and Twitter dataset for scalability reason. We do not include any network embedding methods with contrastive learning and graph neural network as baselines as these methods are typically far from scalable enough for our dataset.

Downstream learners. For fair comparisons, we apply the same downstream learner onto the features derived from each embedding method. Specifically, we use the scikit-learn[5] implementation of logistic regression with l-2 regularization and the SAG solver for both classification and link prediction tasks. Regularization coefficient in the logistic regression is always tuned to the best on the validation set. Following existing research [24], each embedding vector is post-processed by projecting onto the unit l-2 sphere. In multi-label classification tasks, the feature is simply the embedding of each node, and an independent logistic regression model is trained for for each label. In link prediction tasks, features of node pairs are derived by the Hadamard product of the embedding vectors of the two involved node as suggested by previous work [7].

Hyperparameters. For *independent*, MVN2VEC-CON, and MVN2VEC-REG, we set embedding space dimension $D = 128 \cdot |\mathcal{V}|$. For *single-view*, we set $D = 128$. For all other methods, we experiment with both $D = 128$ and $D = 128 \cdot |\mathcal{V}|$, and report the better result. To generate random walk pairs, we set $L = 20$ and $B = 3$. For Snapchat, we set $M = 10$ due to its large scale and set $M = 50$ for all other datasets. The negative sampling rate K is set to be 5 for all models, and each model is trained for 1 epoch.

Metrics. For link prediction tasks, we use two widely used metrics: the area under the receiver operating characteristic curve (ROC-AUC) and the area under the precision-recall curve (PRC-AUC). The receiver operating characteristic curve (ROC) is derived from

[5] http://scikit-learn.org/stable/

Table 4: Mean of link prediction results with standard error in brackets on YouTube and Twitter.

Dataset	YouTube		Twitter	
Metric	ROC-AUC	PRC-AUC	ROC-AUC	PRC-AUC
worst *single-view*	0.831 (0.002)	0.515 (0.004)	0.597 (0.001)	0.296 (0.001)
best *single-view*	0.904 (0.002)	0.678 (0.004)	0.715 (0.001)	0.428 (0.001)
independent	**0.931** (0.001)	**0.745** (0.003)	0.724(0.001)	0.447 (0.001)
one-space	0.914 (0.001)	0.702 (0.004)	0.737 (0.001)	0.466 (0.001)
view-merging	0.912 (0.001)	0.699 (0.004)	0.741 (0.001)	0.469 (0.001)
MVE [21]	0.918 (0.001)	0.714 (0.001)	0.727 (0.001)	0.451 (0.001)
DMNE [14]	0.749 (0.001)	0.311 (0.001)	—	—
MVN2VEC-CON	**0.932** (0.001)	**0.746** (0.001)	0.727 (0.000)	0.453 (0.001)
MVN2VEC-REG	**0.934** (0.001)	**0.754** (0.003)	**0.754** (0.001)	**0.478** (0.001)

Table 5: Mean of link prediction results and of classification results with standard error in brackets on Snapchat.

Dataset	Snapchat			
Task	Link prediction		Classification	
Metric	ROC-AUC	PRC-AUC	ROC-AUC	PRC-AUC
worst *single-view*	0.587 (0.001)	0.675 (0.001)	0.634 (0.001)	0.252 (0.001)
best *single-view*	0.592 (0.001)	0.677 (0.002)	0.667 (0.002)	0.274 (0.002)
independent	0.617 (0.001)	0.700 (0.001)	0.687 (0.001)	0.293 (0.002)
one-space	0.603 (0.001)	0.688 (0.002)	0.675 (0.001)	0.278 (0.001)
view-merging	0.611 (0.001)	0.693 (0.002)	0.672 (0.001)	0.279 (0.001)
MVN2VEC-CON	0.626 (0.001)	0.709 (0.001)	**0.693** (0.001)	**0.298** (0.001)
MVN2VEC-REG	**0.638** (0.001)	**0.712** (0.002)	0.690 (0.001)	0.296 (0.002)

plotting true positive rate against false positive rate as the threshold varies, and the precision-recall curve (PRC) is created by plotting precision against recall as the threshold varies. Higher values are preferable for both metrics. For multi-label classification tasks, we also compute the ROC-AUC and the PRC-AUC for each label and report the mean value averaged across all labels.

6.3 Quantitative Evaluation Results on Real-World Datasets

The main quantitative evaluation results are presented in Table 4 and Table 5. Among the in-house baselines and the proposed MVN2VEC models, *all models leveraging multiple views outperformed those using only one view*, which justifies the necessity of using multi-view networks. Moreover, *one-space and view-merging had comparable performance* on each dataset. This is an expected outcome because they both only model *collaboration* and differ from each other merely in whether random walks are performed across network views.

In case the need for either *collaboration* or *preservation* prevails. On YouTube, the proposed MVN2VEC models perform as good but do not significantly exceed the baseline *independent* model. Recall that the need for *preservation* in the YouTube network is overwhelmingly dominating as discussed in Section 4. As a result, it is not surprising to see that additionally modeling *collaboration* does not bring about significant performance boost in such extreme case. On Twitter, *collaboration* plays a more important role than *preservation*, as confirmed by the better performance of *one-space* than *independent*. Furthermore, MVN2VEC-REG achieved better performance than all baselines, while MVN2VEC-CON outperformed *independent* by further modeling *collaboration*, but failed to exceed *one-space*. This phenomenon can be explained by the fact that $\{\mathbf{f}_u^v\}_{v\in\mathcal{V}}$ in MVN2VEC-CON are set to be independent regardless of its hyperparameter $\theta \in [0, 1]$, and MVN2VEC-CON's capability of modeling *collaboration* is bounded by this design.

In case both *collaboration* and *preservation* are indispensable. The Snapchat network used in our experiments lies in between YouTube and Twitter in terms of the need for *preservation* and *collaboration*. The proposed two MVN2VEC models both outperformed all baselines under all metrics. In other words, this experiment result shows the feasibility of gaining performance boost by simultaneously model *preservation* and *collaboration* without over-complicating the model or adding supervision.

The multi-label classification results on Snapchat are presented in Table 5. As with the previous link prediction results, the two MVN2VEC model both outperformed all baselines under all metrics, with a difference that MVN2VEC-CON performed better in this classification task, while MVN2VEC-REG outperformed better in the previous link prediction task. Overall, while MVN2VEC-CON and MVN2VEC-REG may have different advantages in different tasks, they both outperformed all baselines by simultaneously modeling *preservation* and *collaboration* on the Snapchat network, where both *preservation* and *collaboration* co-exist.

External baselines. MVE underperformed MVN2VEC models on YouTube. We interpret this outcome as MVE is a collaboration framework that does not explicitly model *preservation*, which has been shown to be needed in the YouTube datasets. Meanwhile, MVE did not get performance boost from its attention mechanism since the experiment setting forbids it from consuming informative supervision as discussed in Section 6.2. Without additional optimization, the released DMNE implementation did not finish training on the Twitter dataset after two days on a machine with 40 cores of Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz. The comparatively worse performance of DMNE on YouTube should partially attribute to the use of default hyperparameters as described in Section 6.2. Another possible explanation is that DMNE is designed for the more general multi-networks, not optimized for multi-view networks.

These observations in combine demonstrated the feasibility of gaining performance boost by simultaneously modeling *preservation* and *collaboration* without over-complicating the model or requiring additional supervision.

6.4 Hyperparameter Study

We study the **impact of θ for MVN2VEC-CON and γ for MVN2VEC-REG** in real world datasets, which corroborated the observations in Section 6.3. Additionally, to rule out the possibility that *one-space* could actually preserve the view-specific information as long as the embedding dimension were set to be large enough, we study the **impact of embedding dimension** on synthetic dataset used in the case study. Details of the hyperparameter study are presented in Appendix F.

7 CONCLUSION AND FUTURE WORK

We identified and studied the objectives that are specific and important to multi-view network embedding, *i.e. preservation* and *collaboration*. We then explored the feasibility of better embedding by simultaneously modeling both objectives, and proposed two multi-view network embedding methods. Experiments with various downstream tasks were conducted on three real-world multi-view networks and a series of synthetic networks from distinct sources, including two public datasets and a large-scale internal Snapchat dataset. The results corroborated the validity and importance of *preservation* and *collaboration* as two optimization objectives, and demonstrated the effectiveness of the proposed MVN2VEC methods.

Knowing the existence of the identified objectives, future works include modeling different extent of *preservation* and *collaboration* for different pairs of views in multi-view embedding. It is also rewarding to explore supervised methods for task-specific multi-view network embedding that jointly model *preservation* and *collaboration*.

Acknowledgments. This work was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). Any opinions, findings, and conclusions or recommendations expressed in this document are those of the author(s) and should not be interpreted as the views of any U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] Jon P Alston. 1989. Wa, guanxi, and inhwa: Managerial principles in Japan, China, and Korea. *Business Horizons* 32, 2 (1989).
[2] Sezin Kircali Ata, Yuan Fang, Min Wu, Jiaqi Shi, Chee Keong Kwoh, and Xiaoli Li. 2021. Multi-view collaborative network embedding. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 3 (2021), 1–18.
[3] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation Learning for Attributed Multiplex Heterogeneous Network. In *KDD*.
[4] Ove Frank and Krzysztof Nowicki. 1993. Exploratory statistical analysis of networks. *Annals of Discrete Mathematics* 55 (1993).
[5] Isabella Gollini and Thomas Brendan Murphy. 2014. Joint Modelling of Multiple Network Views. *JCGS* (2014).
[6] Derek Greene and Pádraig Cunningham. 2013. Producing a unified graph representation from multiple social network views. In *Web Science Conference*.
[7] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.
[8] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*. PMLR, 4116–4126.
[9] Abhishek Kumar, Piyush Rai, and Hal Daume. 2011. Co-regularized multi-view spectral clustering. In *NIPS*. 1413–1421.
[10] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.
[11] Weiyi Liu, Pin-Yu Chen, Sailung Yeung, Toyotaro Suzumura, and Lingli Chen. 2017. Principled multilayer network embedding. In *ICDM Workshops*.
[12] Guixiang Ma, Lifang He, Chun-Ta Lu, Weixiang Shao, Philip S Yu, Alex D Leow, and Ann B Ragin. 2017. Multi-view clustering with graph embedding for connectome analysis. In *CIKM*.
[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.
[14] Jingchao Ni, Shiyu Chang, Xiao Liu, Wei Cheng, Haifeng Chen, Dongkuan Xu, and Xiang Zhang. 2018. Co-Regularized Deep Multi-Network Embedding. In *WWW*.
[15] Nianwen Ning, Bin Wu, and Chengcheng Peng. 2018. Representation Learning Based on Influence of Node for Multiplex Network. In *DSC*.
[16] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric Transitivity Preserving Graph Embedding.. In *KDD*.
[17] Philippa Pattison and Stanley Wasserman. 1999. Logit models and logistic regressions for social networks: II. Multivariate relations. *BJMSP* 52, 2 (1999), 169–194.
[18] Jian Pei, Daxin Jiang, and Aidong Zhang. 2005. On mining cross-graph quasi-cliques. In *KDD*. ACM, 228–238.
[19] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.
[20] Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. 2017. Don't Walk, Skip! Online Learning of Multi-scale Network Embeddings. In *ASONAM*.
[21] 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*.
[22] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*.
[23] Vikas Sindhwani and Partha Niyogi. 2005. A co-regularized approach to semi-supervised learning with multiple views. In *ICML Workshop on Learning with Multiple Views*.
[24] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*.
[25] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural Deep Network Embedding. In *KDD*.
[26] Yingheng Wang, Yaosen Min, Xin Chen, and Ji Wu. 2021. Multi-view graph contrastive representation learning for drug-drug interaction prediction. In *Proceedings of the Web Conference 2021*. 2921–2933.
[27] Hao Xiong, Junchi Yan, and Li Pan. 2021. Contrastive Multi-View Multiplex Network Embedding with Applications to Robust Network Alignment. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1913–1923.
[28] Linchuan Xu, Jing Wang, Lifang He, Jiannong Cao, Xiaokai Wei, S Yu Philip, and Kenji Yamanishi. 2019. MixSp: a framework for embedding heterogeneous information networks with arbitrary number of node and edge types. *IEEE Transactions on Knowledge and Data Engineering* 33, 6 (2019), 2627–2639.
[29] R. Zafarani and H. Liu. 2009. Social Computing Data Repository at ASU. http://socialcomputing.asu.edu
[30] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. 2018. Scalable Multiplex Network Embedding.. In *IJCAI*.
[31] Jianan Zhao, Qianlong Wen, Shiyu Sun, Yanfang Ye, and Chuxu Zhang. 2021. Multi-view Self-supervised Heterogeneous Graph Embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 319–334.
[32] Dengyong Zhou and Christopher JC Burges. 2007. Spectral clustering and transductive learning with multiple views. In *ICML*.

APPENDIX

A GRADIENTS OF OBJECTIVE FUNCTIONS

We provide the gradients used for ASGD in the proposed algorithms.

MVN2VEC-CON :

$$\frac{\partial O_{\text{CON}}}{\partial \mathbf{f}_u^v} = \left(1-\sigma(\mathbf{f}_u^v \cdot \varphi_\theta^v(\{\tilde{\mathbf{g}}_n^{v'}\}_{v'\in\mathcal{V}}))\right)\cdot \varphi_\theta^v(\{\tilde{\mathbf{g}}_n^{v'}\}_{v'\in\mathcal{V}}) - \sum_{i=1}^{K}\sigma(\mathbf{f}_u^v \cdot \varphi_\theta^v(\{\tilde{\mathbf{g}}_{n_i'}^{v'}\}_{v'\in\mathcal{V}}))\cdot \varphi_\theta^v(\{\tilde{\mathbf{g}}_{n_i'}^{v'}\}_{v'\in\mathcal{V}}), \tag{8}$$

$$\frac{\partial O_{\text{CON}}}{\partial \tilde{\mathbf{g}}_n^{\hat{v}}} = \left(1-\sigma(\mathbf{f}_u^v \cdot \varphi_\theta^v(\{\tilde{\mathbf{g}}_n^{v'}\}_{v'\in\mathcal{V}}))\right)\cdot \mathbf{f}_u^v \cdot \begin{cases}\theta+\frac{1-\theta}{|\mathcal{V}|}, & \hat{v}=v,\\ \theta, & \hat{v}\neq v,\end{cases} \tag{9}$$

$$\frac{\partial O_{\text{CON}}}{\partial \tilde{\mathbf{g}}_{n_i'}^{\hat{v}}} = -\sigma(\mathbf{f}_u^v \cdot \varphi_\theta^v(\{\tilde{\mathbf{g}}_{n_i'}^{v'}\}_{v'\in\mathcal{V}}))\cdot \mathbf{f}_u^v \cdot \begin{cases}\theta+\frac{1-\theta}{|\mathcal{V}|}, & \hat{v}=v,\\ \theta, & \hat{v}\neq v.\end{cases} \tag{10}$$

MVN2VEC-REG :

$$\frac{\partial O_{\text{REG}}}{\partial \mathbf{f}_u^v} = \left(1-\sigma(\mathbf{f}_u^v \cdot \tilde{\mathbf{f}}_n^v)\right)\cdot \tilde{\mathbf{f}}_n^v - \sum_{i=1}^{K}\sigma(\mathbf{f}_u^v \cdot \tilde{\mathbf{f}}_{n_i'}^v)\cdot \tilde{\mathbf{f}}_{n_i'}^v + 2\gamma\Big(K+1\Big)\Big(1-\frac{1}{|\mathcal{V}|}\Big)\cdot\Big(\mathbf{f}_u^v - \frac{1}{|\mathcal{V}|}\sum_{v'\in\mathcal{V}}\mathbf{f}_u^{v'}\Big), \tag{11}$$

$$\frac{\partial O_{\text{REG}}}{\partial \tilde{\mathbf{f}}_n^v} = \left(1-\sigma(\mathbf{f}_u^v \cdot \tilde{\mathbf{f}}_n^v)\right)\cdot \mathbf{f}_u^v + 2\gamma\Big(1-\frac{1}{|\mathcal{V}|}\Big)\cdot\Big(\tilde{\mathbf{f}}_n^v - \frac{1}{|\mathcal{V}|}\sum_{v'\in\mathcal{V}}\tilde{\mathbf{f}}_n^{v'}\Big), \tag{12}$$

$$\frac{\partial O_{\text{REG}}}{\partial \tilde{\mathbf{f}}_{n_i'}^v} = -\sigma(\mathbf{f}_u^v \cdot \tilde{\mathbf{f}}_{n_i'}^v)\cdot \mathbf{f}_u^v + 2\gamma\Big(1-\frac{1}{|\mathcal{V}|}\Big)\cdot\Big(\tilde{\mathbf{f}}_{n_i'}^v - \frac{1}{|\mathcal{V}|}\sum_{v'\in\mathcal{V}}\tilde{\mathbf{f}}_{n_i'}^{v'}\Big). \tag{13}$$

Note that in implementation, $|V|$ should be the number of views in which u is associated with at least one edge.

B RANDOM WALK PAIR GENERATION

Without additional supervision, we assume the equal importance of different network views and sample the same number $N \in \mathbb{N}$ of random walks from each view. To determine N, we denote $n^{(v)}$ the number of nodes that are not isolated from the rest of the network in view $v \in \mathcal{V}$, $n_{max} := \max\{n^{(v)} : v \in \mathcal{V}\}$, and let $N := M \cdot n_{max}$, where M is a hyperparameter to be specified. Given a network view $v \in \mathcal{V}$, we generate random walk pairs following existing studies [7, 19, 20], where the transition probabilities from a node are proportional to the weights of all its outgoing edges. Specifically, each random walk is of length $L \in \mathbb{N}$, and $\lfloor N/n^{(v)} \rfloor$ or $\lceil N/n^{(v)} \rceil$ random walks are sampled from each non-isolated node in view v, yielding a total of N random walks. For each node in any random walk, this node and any other node within a window of size $B \in \mathbb{N}$ form a random walk pair that is then added to $\mathcal{W}^{(v)}$.

C COMPLEXITY ANALYSIS

For every view, random walks are generated independently by existing method [7, 19, 20]. An analysis similar to that in related work [7] can show the overall complexity is $O\left(\frac{|\mathcal{V}|\cdot L}{B\cdot(L-B)}\right)$, which is linear to the number of views $|\mathcal{V}|$. For model inference, the number of ASGD steps is $|\mathcal{W}|$, linear to L, B, M and $|\mathcal{V}|$. Each ASGD step computes gradients given in the supplementary file, which is again linear to $|\mathcal{V}|$. Therefore, the overall complexity of model inference is quadratic to the number of views $|\mathcal{V}|$.

D THE MVN2VEC ALGORITHMS

We summarize both the MVN2VEC-CON algorithm and the MVN2VEC-REG algorithm in 1.

Algorithm 1: MVN2VEC-CON and MVN2VEC-REG

Input : the multi-view network $G = (\mathcal{U}, \{\mathcal{E}^{(v)}\}_{v\in\mathcal{V}})$ and the hyperparameters
Output: the final embedding $\{\mathbf{f}_u\}_{u\in\mathcal{U}}$
begin
  for $v \in \mathcal{V}$ **do**
    Sample a list $\mathcal{W}^{(v)}$ of random walk pairs
  Join and shuffle the lists of random walk pairs from all views to form a new random walk pair list $\mathcal{W}$
  for *each epoch* **do**
    `// The following for-loop is parallelized`
    for $(u, n) \in \mathcal{W}$ **do**
      if *training* MVN2VEC-CON **then**
        Update $\{\mathbf{f}_u^v, \tilde{\mathbf{g}}_u^v\}_{u\in\mathcal{U},v\in\mathcal{V}}$ with one step descent using gradients in Eq. (8)–(10)
      if *training* MVN2VEC-REG **then**
        Update $\{\mathbf{f}_u^v, \tilde{\mathbf{f}}_u^v\}_{u\in\mathcal{U},v\in\mathcal{V}}$ with one step descent using gradients in Eq. (11)–(13)
  for $u \in \mathcal{U}$ **do**
    Derive the embedding for node u by $\mathbf{f}_u = \bigoplus_{v\in\mathcal{V}} \mathbf{f}_u^v$

E CASE STUDY – VARIED *PRESERVATION* AND *COLLABORATION* ON SYNTHETIC DATA

To directly study the relative performance of different models on networks with varied extent of *preservation* and *collaboration*, we design a series of synthetic networks and conduct a multi-class classification task.

We denote each of these synthetic networks by $G(p)$, where $p \in [0, 0.5]$ is referred to as the intrusion probability. Each $G(p)$ has 4,000 nodes and 2 views – v_1 and v_2. Furthermore, each node is associated to one of the 4 class labels – A, B, C, or D – and each class has exactly 1,000 nodes. Before introducing the more general $G(p)$, we first describe the process for generating $G(0)$ as follows:

(1) generate one random network over all nodes with label A or B, and another over all nodes with label C or D; put all edges in these two random networks into view v_1;
(2) generate one random network over all nodes with label A or C, and another over all nodes with label B or D; put all edges in these two random networks into view v_2.

To generate each of the four aforementioned random networks, we adopt the widely used preferential attachment process with one edge to attach from a new node to existing nodes, where the preferential attachment process is a widely used method for generating networks with power-law degree distribution. With this design, view v_1 carries the information that nodes labeled A or B should be treated differently from nodes labeled C or D, while v_2 reflects that nodes labeled A or C are different from nodes labeled B or D. More generally, $G(p)$ are generated with the following tweak from $G(0)$:

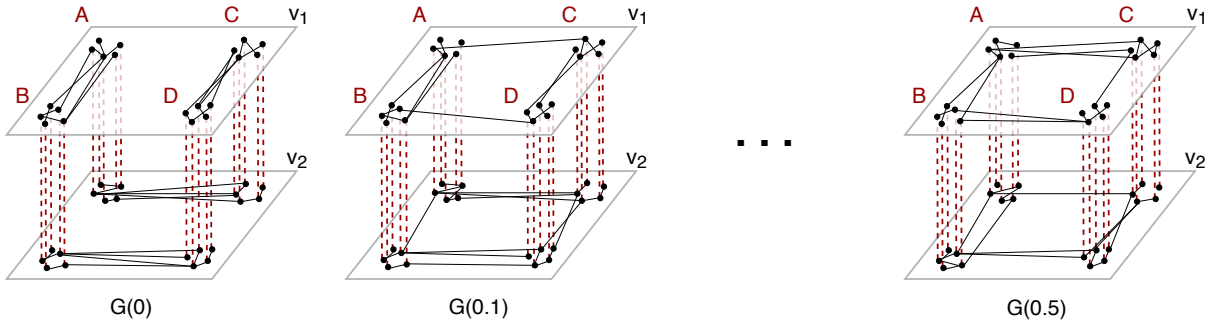


Figure 4: An illustration of the series of synthetic networks $G(p)$.

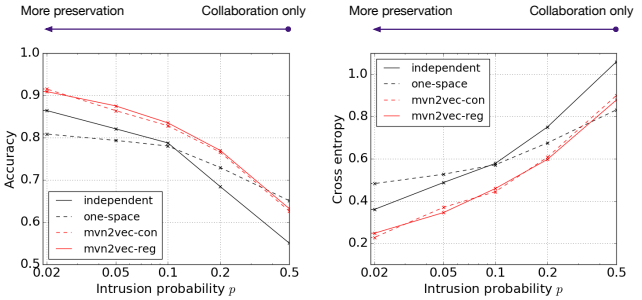


Figure 5: Classification results under accuracy and cross entropy on synthetic networks $G(p)$ with varied intrusion probability p, corresponding to different extent of *preservation* and *collaboration*.

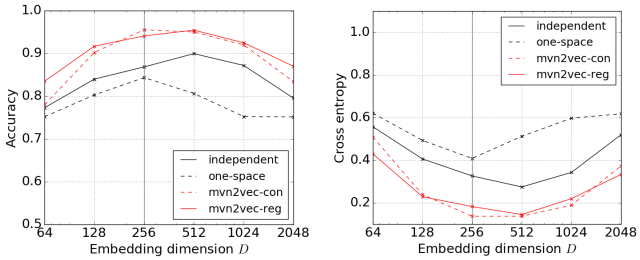


Figure 6: Classification results under accuracy and cross entropy on network $G(0)$ with varied embedding dimension D.

when putting an edge into one of the two views, with probability p, the edge is put into the other view instead of the view specified in the $G(0)$ generation process.

It is worth noting that greater p favors more *collaboration*, while smaller p favors more *preservation*. In the extreme case where $p = 0.5$, only *collaboration* is needed in the network embedding process. This is because every edge has equal probability to fall into view v_1 or view v_2 of $G(0.5)$, and there is hence no information carried specifically by either view that should be preserved.

For each $G(p)$, *independent*, *one-space*, MVN2VEC-CON, and MVN2VEC-REG are tested. Atop the embedding learned by each model, we apply logistic regression with cross-entropy to carry out the multi-class evaluation tasks. Parameters are tuned on a validation dataset sampled from the 4,000 class labels. Classification accuracy and cross-entropy on a different test dataset are reported in Figure 5.

Observations. From Figure 5, we make three observations. (i) *independent* performs better than *one-space* in case p is small – when *preservation* is the dominating objective in the network – and *one-space* performs better than *independent* in case p is large – when *collaboration* is dominating. (ii) The two proposed MVN2VEC models perform better than both *independent* and *one-space* except when p is close to 0.5, which implies it is indeed possible for MVN2VEC to achieve better performance by simultaneously modeling *preservation* and *collaboration*. (iii) When p is close to 0.5, *one-space* performs the best. This is expected because no *preservation* is needed in $G(0.5)$, and any attempts to additionally model *preservation* shall not boost, if not impair, the performance.

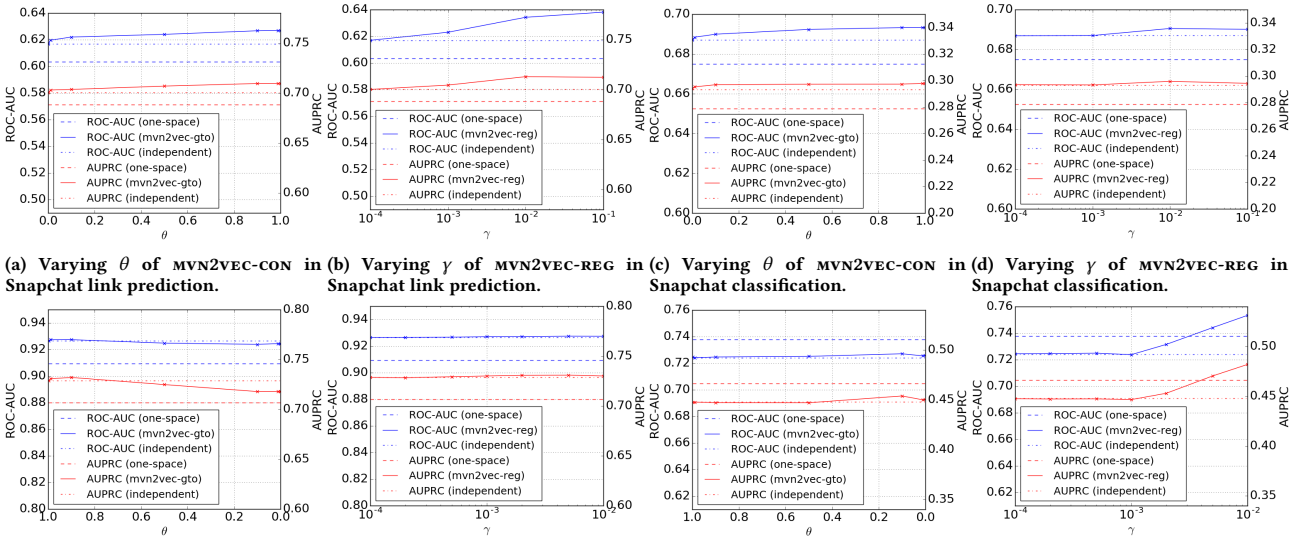
F HYPERPARAMETER STUDY

Impact of θ for MVN2VEC-CON and γ for MVN2VEC-REG. We study the impact of θ for MVN2VEC-CON and γ for MVN2VEC-REG, which corroborated the observations in Section 6.3. With results presented in Figure 7, we first focus on the Snapchat network. Starting from $\gamma = 0$, where only *preservation* was modeled, MVN2VEC-REG performed progressively better as more *collaboration* kicked in by increasing γ. The peak performance was reached between 0.01 and 0.1. On the other hand, the performance of MVN2VEC-CON improved as θ grew. Recall that even in case $\theta = 1$, MVN2VEC-CON still have $\{\mathbf{f}_u^v\}_{v \in \mathcal{V}}$ independent in each view. This prevented MVN2VEC-CON from promoting more *collaboration*.

On YouTube, the MVN2VEC models did not significantly outperform *independent* no matter how θ and γ varied due to the dominant need for *preservation* as discussed in Section 4 and 6.3.

On Twitter, MVN2VEC-REG outperformed *one-space* when γ was large, while MVN2VEC-CON could not beat *one-space* for reasons discussed in Section 6.3. This also echoed MVN2VEC-CON's performance on Snapchat as discussed in the first paragraph of this section.

Impact of embedding dimension. To rule out the possibility that *one-space* could actually preserve the view-specific information as long as the embedding dimension were set to be large enough, we study the impact of embedding dimension. Particularly, we carry out the multi-class classification task on $G(0)$ under varied embedding



(a) Varying θ of MVN2VEC-CON in Snapchat link prediction. **(b) Varying γ of MVN2VEC-REG in Snapchat link prediction.** **(c) Varying θ of MVN2VEC-CON in Snapchat classification.** **(d) Varying γ of MVN2VEC-REG in Snapchat classification.**

(e) Varying θ of MVN2VEC-CON in YouTube link prediction. **(f) Varying γ of MVN2VEC-REG in YouTube link prediction.** **(g) Varying θ of MVN2VEC-CON in Twitter link prediction.** **(h) Varying γ of MVN2VEC-REG in Twitter link prediction.**

Figure 7: Performance of the MVN2VEC models under varying hyperparameters regarding *preservation* and *collaboration*.

dimensions. Note that $G(0)$ is used in this experiment because it has the need for modeling *preservation* as discussed in Section E. As presented in Figure 6, *one-space* achieves its best performance at $D = 256$, which is worse than *independent* at $D = 256$, let alone the best performance of *independent* at $D = 512$. Therefore, one cannot expect *one-space* to preserve the information carried by different views by employing embedding space with large enough dimension.

Besides, all four models achieve their best performance with D around 256~512. Particularly, *one-space* uses the smallest embedding dimension to reach peak performance. This is expected since *one-space* does not divide the embedding space to suit multiple views and have more freedom in exploiting an embedding space with given dimension.