Multi-view Denoising Graph Auto-Encoders on Heterogeneous Information Networks for Cold-start Recommendation

Jiawei Zheng[1,2], Qianli Ma[1†], Hao Gu[2], Zhenjing Zheng[1]
[1]School of Computer Science and Engineering, South China University of Technology
[2]WeChat Technical Architecture Department, Tencent Inc.
csjwzheng@foxmail.com,qianlima@scut.edu.cn

ABSTRACT

Cold-start recommendation is a challenging problem due to the lack of user-item interactions. Recently, heterogeneous information network (HIN)-based recommendation methods use rich auxiliary information to enhance users and items' connections, helping alleviate the cold-start problem. Despite progress, most existing methods model HINs under traditional supervised learning settings, ignoring the gaps between training and inference procedures in cold-start scenarios. In this paper, we regard cold-start recommendation as a missing data problem where some user-item interaction data are missing. Inspired by denoising auto-encoders that train a model to reconstruct the input from its corrupted version, we propose a novel model called Multi-view Denoising Graph Auto-Encoders (MvDGAE) on HINS. Specifically, we first extract multifaceted meaningful semantics on HINs as multi-views for both users and items, effectively enhancing user/item relationships on different aspects. Then we conduct the training procedure by randomly dropping out some user-item interactions in the encoder while forcing the decoder to use these limited views to recover the full views, including the missing ones. In this way, the complementary representations for both users and items are more informative and robust to adjust to cold-start scenarios. Moreover, the decoder's reconstruction goals are multi-view user-user and item-item relationship graphs rather than the original input graphs, which make the features of similar users (or items) in the meta-paths closer together. Finally, we adopt a Bayesian task weight learner to balance multi-view graph reconstruction objectives automatically. Extensive experiments on both public benchmark datasets and a large-scale industry dataset *WeChat Channel* demonstrate that MvDGAE significantly outperforms the state-of-the-art recommendation models in various cold-start scenarios. The case studies also illustrate that MvDGAE has potentially good interpretability.

CCS CONCEPTS

• **Information systems** → **Data mining**; **Collaborative filtering**.

† The corresponding author.

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467427

KEYWORDS

Multi-view, denoising Graph Auto-Encoder, Heterogeneous Information Network, Cold-start Recommendation

ACM Reference Format:
Jiawei Zheng[1,2], Qianli Ma[1†], Hao Gu[2], Zhenjing Zheng[1] . 2021. Multi-view Denoising Graph Auto-Encoders on Heterogeneous Information Networks for Cold-start Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467427

1 INTRODUCTION

With the explosive growth of online information in e-commerce and social media platforms, recommender systems play a key role in addressing information overload for users. It effectively helps users discover interesting products or contents. However, when the user-item interaction data are sparse, it is challenging to learn effective user or item representations, which leads to the so-called cold-start problem.

Traditional methods deal with the cold-start problem by incorporating user or item content data, such as review texts or images to enhance user and item representations. Despite progress, these methods heavily rely on the availability and quality of content data. The learned model may recommend the same items for users with similar content, thereby neglecting personal interests [2]. There are also some attempts [2, 13, 16] to introduce meta-learning [24] into the recommender systems to solve the cold-start problem from the model level, i.e., predicting a user's preferences with only a few past item interactions. Although these methods achieve promising performance, most of them construct learning tasks aimed at producing personalized parameter initializations for new users, ignoring the problem of providing meaningful initialization representations for new items. Hence, users cannot get recommendations for the new items even though they may be interested. Therefore, it is crucial to generate meaningful initialization representations for new items.

Recently, heterogeneous information network (HIN)-based [22] recommendation methods employ rich auxiliary information to enhance the connections of users and items, which helps overcome the sparsity issue and alleviates the cold-start problem. In Fig. 1, we show a toy example for movie recommendations characterized by HINs. We observe that HINs can capture how the users/movies are related to each other via some auxiliary nodes, such as actors, in addition to the existing user-movie interactions. For example, the meta-path [23] User-Movie-User can characterize users' interest similarity to some extent, and Movie-Actor-Movie can enhance the

similarity of movies played by the same actor. The HIN-based recommendation has received much attention and made great progress in the literature [5, 21, 33]. However, these methods model HINs under the traditional supervised learning setting and ignore the gaps between training and inference procedures in cold-start scenarios.

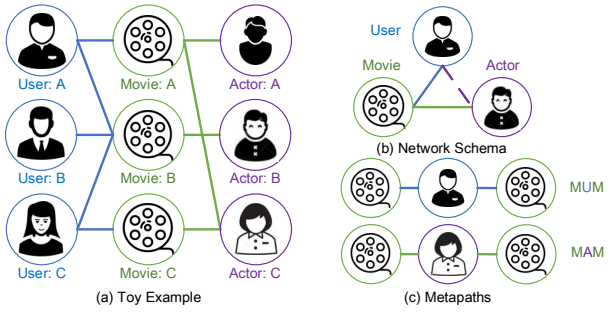

Figure 1: A toy example for movie recommendation characterized by HINs.

In this paper, we regard cold-start as a missing data problem where user-item interaction data is missing. Inspired by denoising auto-encoders [26] that train the model to reconstruct the input from a corrupted version, as well as the classical framework graph auto-encoders [4, 20, 31], we propose a novel model named Multi-view Denoising Graph Auto-Encoders (MvDGAE) on HINS to alleviate the cold-start problem. We first extract multifaceted meaningful semantics reflected by meta-path on HINs as multi-views for both users and items, effectively enhancing user/item relationships on different aspects. For each view, we propose an independent graph encoder network to learn semantic embeddings based on meta-paths. They are then aggregated to form complementary representations. In the training procedure, we randomly drop out some user-item interaction views in the encoder while forcing the decoder to use the limited views to recover the full view. In this way, the complementary representations for both users and items are more informative and robust to adjust to cold-start scenarios. Note that the decoder's reconstruction goals are multi-view user-user and item-item relationship graphs rather than the original input graphs, which makes the features of similar users (or items) in the meta-paths closer together. Finally, we adopt a Bayesian task weight learner to balance multi-view graph reconstruction objectives automatically.

The major contributions of this paper summarized are as follows:

- We regard the cold-start as a missing data problem and propose a denoise graph auto-encoder framework. The training randomly drops out some user-item interaction views in the encoders but forces the decoders to reconstruct the full view information, significantly alleviating the cold-start problem.
- Instead of rebuilding the original input graphs by the decoders, our reconstruction goals are multi-view user-user and item-item relationship graphs balanced by Bayesian task weight learner automatically, which make the features of the similar users (or items) in the meta-paths more close together.
- Extensive experiments on both public benchmarks and real industry large-scale datasets *WeChat Channel* demonstrate that MvDGAE significantly outperforms the state-of-the-art recommendation models in different cold-start scenarios. The case studies also illustrate the good interpretability of our model.

2 RELATED WORK

2.1 Cold-start Recommendation

In the recommendation system, when the user-item interaction data are sparse, it is difficult to learn effective user or item representations, which leads to the so-called cold-start problem. Traditional methods rely on feature engineering to deal with the cold-start problem by incorporating content data. Additionally, some transfer learning-based methods [10, 14] are also adopted to alleviate the cold-start problem, which uses the well-learned representations of the overlapped objects from the source domain to the target domain. Although the user and item representations are enhanced by these content-based features or extreme domains, it heavily relies on the availability and quality of them. There are also some attempts [2, 13, 16] introduce meta-learning [24] into the recommender systems to solve the cold-start problem from the model-level, i.e., predicting a user's preferences by only a few past interacted items. The core idea of meta-learning based recommendation methods is learning a global parameter to initialize the parameter of personalized recommender models, which are further locally updated to learn a specific user's preference. But most of them construct learning tasks based on users aiming to produce a personalized parameter initialization for new users instead of for new items. Recently, (HIN)-based recommendation methods [5, 21, 33] use rich auxiliary information to enhance the connections of users and items, which help to overcome the sparsity issue and alleviate the cold-start problem. However, these methods model HINs under traditional supervised learning settings, ignoring the gaps between training and inference procedure in cold-start scenarios.

We regard the cold-start as a missing data problem and propose a denoise graph auto-encoder framework. In the training process, we randomly drop out some user-item interaction views in the encoders but forces the decoders to reconstruct the full view information, significantly alleviating the cold-start problem.

2.2 Heterogeneous Information Network Embedding

HINs (Heterogeneous Information Network) have been proposed to model complex objects and their rich relations. HIN embedding aims to embed the nodes in the network into a low-dimensional space, which help to facilitate downstream applications such as link prediction, personalized recommendation, node classification, etc [22, 30]. One line of work leverages meta-path-based contexts for semantic-preserving embedding. For example, Dong et al. [3] introduced meta path guided walks, skip-gram model [17] and negative sampling to learn heterogeneous representations. HIN2Vec [6] combined various prediction training tasks jointly based on a target set of relationships to learn embedding of nodes and meta-paths in the HIN. Note that these methods rely on domain knowledge to choose the right meta-paths [9]. Another line adopts the Deep Neural Networks, such as MLP, Auto-encoder, GNN, etc., to model heterogeneous data. SHINE [28] utilizes multiple deep auto-encoders

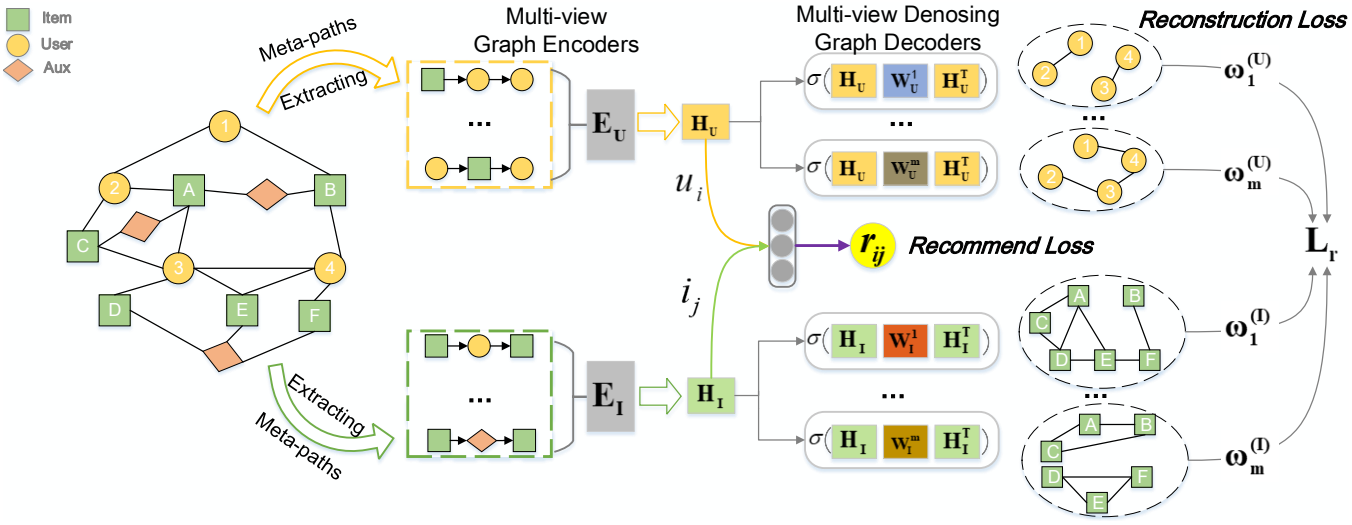

Figure 2: The framework of MvDGAE.

to map each node into a low-dimension space while preserving the heterogeneous network structure. HetGNN [32] proposed a heterogeneous graph neural network to aggregate feature information of sampled neighboring nodes to obtain the ultimate node embedding. HAN [29] introduced both of the node-level and semantic-level attentions to model the importance of nodes and meta-paths simultaneously and thus generate node embedding in a hierarchical manner.

3 PRELIMINARIES

In this section, we introduce relevant concepts used in this paper, including Heterogeneous Information Network, Network Schema, Meta-path, and Cold-start Recommendation Problem.

DEFINITION 1. **Heterogeneous Information Network (HIN).** An HIN is defined as a graph $G = (V, E, A, R, \phi, \varphi)$, where V and E are the sets of nodes and edges, respectively. Each node v and edge e are associated with their type mapping functions $\phi : V \rightarrow A$ and $\varphi : E \rightarrow R$, where A and R denote the sets of node and edge types such that $|A| + |R| > 2$.

For better understanding the complex HINs, the **network schema** have been proposed to present the meta structure of a network, including the object types and their interaction relations.

DEFINITION 2. **Network Schema**. The Network Schema is denoted as $S = (A, R)$. It is a meta template for an HIN $G = (V, E)$ with the object type mapping $\phi : V \rightarrow A$ and the edge type mapping $\varphi : E \rightarrow R$. Fig. 1 shows a toy example for movie recommendation characterized by HINs. Its corresponding network schema is shown in Fig. 1(b), consisting of multiple types of objects, including User (U), Movie (M), Actor (A).

Two objects in the HIN can be connected through different semantic paths, which are defined as meta-paths:

DEFINITION 3. **Meta-path**[23]. Given an HIN $G = (V, E, A, R, \phi, \varphi)$, a meta-path ρ is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$, which describes a composite relation between v_1 and v_l. Taking Fig. 1(c) as an example, two objects in the HIN can be connected via multiple meta-paths, e.g., "Movie-User-Movie (MUM)" and "Movie-Actor-Moive (MAM)". Different meta-paths usually convey different semantics.

DEFINITION 4. **Cold-start Recommendation Problem**. On an HIN $G = (V, E, A, R)$, let $V_U, V_I \in V$ denote user and item nodes, respectively. Given the user-item interaction matrix $R = \{r_{u,i} : u \in V_U, i \in V_I, \langle u, i \rangle \in E\}$ and the corresponding HIN G in the system, we aim to predict the unknown rating between user u and item i. Note that if u is a new user with only limited existing ratings, i.e., $\{|r_{\grave{u},i} \in R : \grave{u} = u|\}$ is small or even zero, it is known as **user cold-start (UC)** Recommend; similarly, if i is a new item, it is known as **item cold-start (IC)**; if both u and i are new objects, it is known as **user-item cold-start (UIC)**.

4 PROPOSED METHOD

4.1 Overview of MvDGAE

The basic idea of the proposed model Multi-view Denoising Graph Auto-Encoders (MvDGAE) is to learn meaningful and robust representations for both users and items, which is able to adjust to both warm and cold start scenarios.

The framework of MvDGAE is illustrated in Fig. 2. First, we extract multifaceted meaningful semantics on HINs as multi views for both users and items, effectively enhancing user/item relationships on different aspects. For each view, we propose an independent graph encoder network to learn semantic embeddings based on meta-paths. They are then aggregated to form complementary representations. In the training procedure, we randomly drop out some user-item interaction views in the encoder while forcing the decoder to use the limited views to recover the full view. Note that the decoder's reconstruction goals are multi-view user-user and item-item relationship graphs rather than the original input graphs, which makes the features of similar users (or items)in the meta-paths closer together. Finally, we adopt a Bayesian task weight learner to balance multi-view graph reconstruction objectives automatically.

4.2 Multi-view Graph Encoders

Each type of meta-path can be regarded as a view. In our multi-view denoising graph encoders, there are independent encoder networks instantiated as meta path-guided heterogeneous GNN [5] for each view, and N-layer fully connected networks with shared parameters. Firstly, based on each meta-path, we perform the heterogeneous GNN to capture the rich semantic information with node-level attention, which distinguishes each neighbor's importance. Secondly, we aggregate all the view embeddings with the dropout operation to get more robust complementary representations for cold-start recommendation. The details are as follows.

4.2.1 Node-level Aggregation based on Meta-path. Each neighbor node plays a different role and show different importance for the center node. So we leverage graph attention network (GAT) [25] as the aggregation mechanism to aggregate the neighbors representation with different importance to form a node embedding.

Here we let the node i as the center node and $\mathcal{N}_i$ as its neighbors set to simply illustrate how the GAT work. Due to the heterogeneity of nodes, different types of nodes have different feature spaces [29]. So we first project them to the same node space and then calculate the attention score.

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}\left[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}\left[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_k\right]\right)\right)}, \forall j \in \mathcal{N}_i, \quad (1)$$

where $\boldsymbol{W} \in \mathbb{R}^{F'\times F}$ and $\boldsymbol{a} \in \mathbb{R}^{2F'\times 1}$ denote linear transformation. T represents transposition and || is the concatenation operation. $\mathcal{N}_i$ denotes the neighbors set of node i. We apply LeakyReLU with negative input slope $\alpha = 0.2$ as activation function. Then we aggregate information from $\mathcal{N}_i$:

$$\boldsymbol{h}_i^{'} = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\boldsymbol{h}_j + \boldsymbol{h}_i\right), \quad (2)$$

where $\sigma(\cdot)$ denotes the activation function.

As shown in Fig 3, there are multiple meta-paths for the target user u_e. We take the meta-path UUI as an example to illustrate how to obtain the target user embedding via node-level aggregation.

Firstly, we sample the 1-st step neighbors of u_e, $\mathcal{N}^1_{UUI}(u_e) = \{u_1, u_2\}$. Based on each node u_k in the neighbors set $\mathcal{N}^1_{UUI}(u_e)$, we further sample the 2-nd step neighbors set $\mathcal{N}^2_{UUI}(u_e) = \{i_1, i_2, i_3\}$. After we obtain the sampled 1-st and 2-nd step neighbors set of the target user u_e, we aggregate the embedding of 2-nd step neighbors to obtain the 1-st step neighbors' embeddings and further get the target user u_e embedding. Note that we adopt GAT as the aggregate mechanism to distinguish the neighbors' importance. Following this process, we can obtain the aggregated embedding $\boldsymbol{h}_e^{UUI}$ of target user u_e guided by the meta-path UUI. It reflects that one user may prefer the items that his/her friends are interested in.

4.2.2 Dropout on Multi-views. After obtaining each view embeddings, we then fuse them to get a complementary representation.

$$\boldsymbol{h}_e = \frac{1}{M}\sum_{m=1}^{M}\text{MLP}(\boldsymbol{h}_e^m), \quad (3)$$

where m denotes the m-th view and M is the number of views.

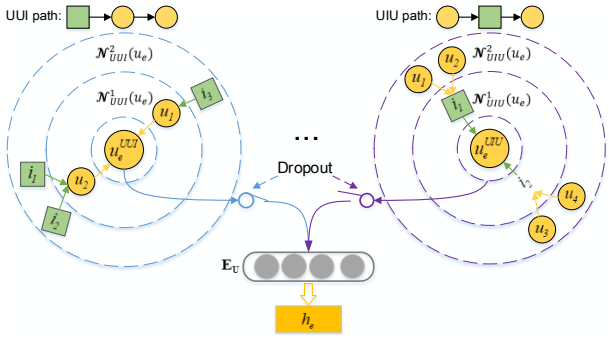

Figure 3: The Multi-view Graph Encoders architecture of MvDGAE.

During training, we aim to generalize the model to adjust to the cold start scenario. We are inspired by denoising auto-encoders [26] whose goal is to still produce accurate representations when parts of the input are missing. Specifically, we randomly drop out some views based on the user-item interaction, such as UIU, IUI, etc. Then the complementary representations in Eq. 3 are modified as:

$$\boldsymbol{h}_e = \frac{1}{M}\sum_{m=1}^{M}\text{MLP}(\boldsymbol{h}_e^m) * S(m), \quad (4)$$

where $S(m) \in \{0, 1\}$ indicates that whether m-th view is dropped out or not.

This training strategy with dropout has two advantages. On the one hand, multi-view graph encoders with dropout encourage the model to pay more attention to capture the informative features from helpful auxiliary views. In case of missing the user-item interaction views, it is also capable of producing meaningful representations with more robustness. In this way, the multi-view graph encoders can generalize to the cold start scenario naturally. On the other hand, dropout can be used as an effective way of regularizing the model to avoid over-fitting.

4.3 Multi-view Graph Denoising Decoding

To make the features of similar users (or items) in the meta-paths closer together, the complementary representations are further required to reconstruct multi-view user-user and item-item relationship graphs rather than the original input graphs, including the missing one.

4.3.1 Construct Multi-View Graph. In order to better model the representation similarity of users and items in recommendation, motivated by [33], we transform the original auxiliary relationships to user-user relationships or item-item relationships. Concretely, we utilize the second-order proximity [7] to capture the similarity between two users (or items). In this way, we can translate the semantic information brought by auxiliary nodes into user-user relationships and/or item-item relationships and thus to construct the corresponding graphs. Benefit from rich semantic information in HINs, we can obtain multi-view graphs to directly describe user-user and item-item relationships.

4.3.2 Multi-View Graph Decoding. To make the fusion representation maintain the characteristics of each type meta-path, it requires to reconstruct the multi-view graph data $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(M)}$ from the fusion representation $\boldsymbol{H}$. Note that $\boldsymbol{H}$ is an embedding matrix where each row represents the node embedding. And for simplicity, here we don't distinguish user/item representations.

As shown in the **Multi-View Graph Decoding** part of Fig 2, each view has its corresponding decoders $\left\{p\left(\hat{\boldsymbol{A}}^{(m)} \mid \boldsymbol{H}, \boldsymbol{W}_m\right)\right\}_{m=1}^{M}$, which aims to predict whether there is a link between two nodes in view m, where $\boldsymbol{W}_m \in \mathbb{R}^{D \times D}$ is the view-specific weights for view m and D is the embedding dimension. Specifically, we reconstruct multi-view graph based on the fusion representation:

$$\sum_{m=1}^{M} p\left(\hat{\boldsymbol{A}}^{(m)} \mid \boldsymbol{H}, \boldsymbol{W}_m\right) = \sum_{m=1}^{M} \text{sigmoid}\left(\boldsymbol{H} \cdot \boldsymbol{W}_m \cdot \boldsymbol{H}^{\mathrm{T}}\right). \quad (5)$$

Here we treat the reconstruction as the binary classification task, i.e., predict whether there is a link between two nodes. Formally, we use the binary cross entropy loss as the reconstruction loss:

$$\begin{aligned} L_r &= \sum_{m=1}^{M} L_r^{(m)} = \sum_{m=1}^{M} \text{loss}\left(\boldsymbol{A}^{(m)}, \hat{\boldsymbol{A}}^{(m)}\right) \\ &= -\sum_{m=1}^{M} \frac{1}{Q^{(m)}} \sum_{(i,j) \in \boldsymbol{A}^{(m)}} \left(a_{ij}^{(m)} \log \hat{a}_{ij}^{(m)} \right. \\ &\quad \left. + \left(1 - a_{ij}^{(m)}\right) \log\left(1 - \hat{a}_{ij}^{(m)}\right)\right), \end{aligned} \quad (6)$$

where $a_{ij} = 1$ if there is a link between node i and j else $a_{ij} = 0$. $Q^{(m)}$ denotes the elements number of matrix $\mathrm{A}^{(m)}$.

In this way, benefit from the multi-view decoding tasks, the gradients will propagate through the graph encoder during the backpropagation process. Therefore, when forward-propagation is processed, the graph encoder will extract the complementary representations by all views.

4.3.3 Sampling Strategy. As mention above, the additional user-user and item-item relationships are generated according to the second order proximity via auxiliary nodes. However, in large-scale graph, the computation of second order proximity in all the nodes is not feasible, which results in $O(N^2)$ and N is the number of nodes. In addition, when we use a mini-batch training, the connection of user-user or item-item in the constructed graph will be very sparse.

Instead of computing second order proximity in all the nodes, we only calculate it in a mini-batch. By this way, the complexity is reduced $O(N^2)$ to $O(\frac{N}{n} \times n^2) = O(N \times n)$, where $n \ll N$ denotes the batch size. Meanwhile, to guarantee the density of the constructed graph, we first sample the auxiliary nodes and then based on it to sample the corresponding target nodes(i.e., user/item nodes) to guarantee the density of the constructed graph.

4.4 Bayesian Task Weight Learner

Multi-view graph decoding can also be viewed as multi-task learning. Each task aims to reconstruct the specific constructed graph. Considering both user and item aspects, the total reconstruction loss are defined as:

$$\mathcal{L}_r = \sum_j w_j \cdot L_{rj}^{(u)} + \sum_k w_k \cdot L_{rk}^{(i)}, \quad (7)$$

where (u) and (i) denote the users and items, respectively. w_j and w_k are hyper-parameters that balance the importance of different view reconstruction objectives.

However, manually tuning these hyper-parameters is expensive and intractable. Instead, inspired by the recent study which uses uncertainty to weigh losses in multi-task learning [11], we leverage a Bayesian task weight learner that can automatically achieve the balance among multi-tasks.

Due to space limit, here we only show the derived result and put the detailed derivation process in the Supplement A.5. Benefiting from the Bayesian Task Weight Learner, each view of user and item is assigned a learnable weight to automatically achieve the balance among multi views. Thus, the total reconstruction loss in Eq. 7 can be formulated as:

$$\begin{aligned} \mathcal{L}_r &= \sum_j -\log \Pr(\boldsymbol{A} \mid \hat{\boldsymbol{A}}, w_j) + \sum_k -\log \Pr(\boldsymbol{A} \mid \hat{\boldsymbol{A}}, w_k) \\ &= \sum_j \left(\frac{1}{w_j^2} L_{rj}^{(u)} + 2 \cdot \log w_j^2\right) + \sum_k \left(\frac{1}{w_k^2} L_{rk}^{(i)} + 2 \cdot \log w_k^2\right), \end{aligned} \quad (8)$$

where w_j and w_k are automatically learned during optimization.

4.5 Optimization Objective

Through aggregating all the views information of users and items, we obtain the fused user embedding $\boldsymbol{h}_e^{(u)}$ for user u_e and item embedding $\boldsymbol{h}_j^{(i)}$ for item i_j. Then we concatenate the fused embeddings of user and item and feed into an *MLP* to get the predict score $\hat{r_{ej}}$. We have:

$$\hat{r}_{ej} = sigmoid(f(\boldsymbol{h}_e^{(u)} \oplus \boldsymbol{h}_j^{(i)})), \quad (9)$$

where (u) and (i) denote the users and items, respectively. The $f(\cdot)$ is the MLP, $sigmoid(\cdot)$ denotes the sigmoid activation, and $\oplus$ is the concatenate operation.

Finally, we minimize the Mean Squared Error (MSE) loss to learn the user preferences:

$$\mathcal{L}_{rec} = \frac{1}{|\mathcal{R}|} \sum_{(e,j) \in \mathcal{R}} \left(r_{ej} - \hat{r}_{ej}\right)^2, \quad (10)$$

where $\mathcal{R}$ denotes the user-item interaction matrix, and r_{ej} is the actual rating of user u_e on item i_j.

Overall objective function. We jointly optimize the Multi-View Graph Auto-Encoder embedding and recommendation learning, and the total objective function is defined as:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_{rec}, \quad (11)$$

where $\mathcal{L}_r$ is the multi-view reconstruction loss and $\mathcal{L}_{rec}$ is the recommendation prediction loss.

5 EXPERIMENTS

5.1 Experimental Setup

Dataset. We evaluate our proposed method on three public benchmark datasets, i.e., DBook [1], MovieLens [2] and Yelp [3], which are provided by [16], including the training/testing data split. In addition, we collect a real-world large-scale dataset from WeChat Channel [4]. The datasets statistics are summarized in Table 3 in Supplement A.2.

For each dataset, users and items are divided into two groups: existing and new. The training data only contains existing users interacting with existing items. The rest are testing data, which are divided into four situations: (1) User Cold-start (**UC**): recommendation of existing items for new users; (2) Item Cold-start (**IC**): recommendation of new items for existing users; (3) User-Item Cold-Start (**UIC**): recommendation of new items for new users; (4) **Normal**: recommendation of existing items for existing users.

Baselines. To validate the effectiveness of our proposed model, we compare MvGAE with three categories of methods: **(1) Traditional methods**, including FM [19], NeuMF [8], DropoutNet [27], and GC-MC [1]. As they are not designed for HINs, following [16], we take the heterogeneous information as the features of users or items. **(2) Meta-learning based methods**, including MeteEmb [18], MeLU [13] and MetaHIN [16]. **(3) HIN-based methods**, including metapath2vec [3], HERec [21], HAN [29]. More detailed introduction and implementation of baselines are included in Supplement A.3.

Evaluation Metrics. As for three public benchmark datasets, where users rate items from 1 to 5, we adopt three widely-used evaluation protocols as [16], namely, mean absolute error (MAE), root mean square error (RMSE), and normalized discounted cumulative gain at rank K (NDCG@K). Here we use $K = 5$. As for WeChat Channel dataset, there are only users' like history without the rating for items. In this case, we use Area Under receiver operator characteristic Curve (AUC) [15] to evaluate the performance. The larger AUC value means better performance.

5.2 Performance Comparison

In this section, we compare MvDGAE to several state-of-the-art baselines, including three cold-start situations (UC, IC, UIC) and traditional non-cold start scenario (Normal). The comparison results on public benchmark datasets and *WeChat Channel* dataset are shown in Table 1 and Table 2, respectively.

Cold-start Scenario. The first three parts of Table 1 and Table 2 present three cold-start scenarios (UC, IC and UIC). Generally, MvDGAE achieve the best performance among all methods. Among different type of baselines, the performance of traditional methods are least competitive. The main reason is that the heterogeneous information are incorporated as content features, ignoring the meaningful semantics in HINs. As for Meta-learning based methods, they show the great performance owing to the well-designed training strategy for producing a personalized parameter initialization for new users. However, they still underperform the proposed MvDGAE in all scenarios. We attribute it to the overlooks on the meaningful initialization for new items. HIN-based methods also perform well benefiting from employing rich auxiliary information to enhance the connections of users and item. Nevertheless, these methods model HINs under traditional supervised learning settings, ignoring the gaps between training and inference procedure in cold-start scenarios.

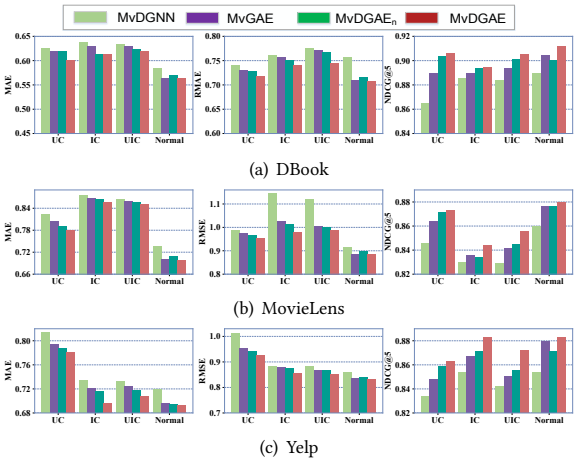

Figure 4: The performance comparison of MvDGAE and its variants. For MAE and RMSE, the smaller, the better. For NDCG@5, the larger, the better.

In contrast, MvDGAE extract multifaceted meaningful semantics on HINs as multi views for both users and items, which effectively enhance user/item relationships on different aspects. The well-designed multi-view denoising graph auto-encoders enable the new users and items to obtain informative and robust representations.

Non-cold-start Scenario. In the last part of Table 1 and Table 2, we report the comparison results on the traditional (normal) recommendation scenario. The proposed MvDGAE still achieves the best performance compared with all the baselines. We contribute these outstanding results to the following reasons. (1) Rich semantics on HINs are modeled as multi views for both users and items to enrich their connections. (2) The complementary representations are required to reconstruct multi-view user-user and item-item relationship graphs rather than the original input graphs, which make the features of similar users (or items) in the meta-paths closer together. (3) Noise on the multi-view graph encoders make the representations more robust.

5.3 Ablation Study

In order to investigate the contribution of each component to the final recommendation performance, we design three variants of MvDGAE: (1) **MvDGNN**: a variant of MvDGAE which only preserve the multi-view graph encoders without the **multi-view**

[1] https://book.douban.com
[2] https://grouplens.org/datasets/movielens/
[3] https://www.yelp.com/dataset/challenge
[4] WeChat Channel is a micro-video sharing platform that allows users to create and share micro-videos. Note that we anonymize the data and conduct strict desensitization processing.

Table 1: Experimental results in four recommendation scenarios and on three public datasets. The best results of all methods are indicated in bold, and second bests are underlined.

Scenario	Model	DBook			MovieLens			Yelp		
		MAE ↓	RMSE ↓	NDCG@5 ↑	MAE ↓	RMSE ↓	NDCG@5 ↑	MAE ↓	RMSE ↓	NDCG@5 ↑
Existing items for new users (User Cold-start or UC)	FM	0.7027	0.9158	0.8032	1.0421	1.3236	0.7303	0.9581	1.2177	0.8075
	NeuMF	0.6541	0.8058	0.8225	0.8569	1.0508	0.7708	0.9413	1.1546	0.7689
	DropoutNet	0.8311	0.9016	0.8114	0.9291	1.1721	0.7705	0.8557	1.0369	0.7959
	GC-MC	0.9061	0.9767	0.7821	1.1513	1.3742	0.7213	0.9321	1.1104	0.8034
	MetaEmb	0.6782	0.8553	0.8527	0.8261	1.0308	0.7795	0.8988	1.0496	0.7875
	MeLU	0.6353	0.7733	0.8793	0.8104	0.9756	0.8415	0.8341	1.0017	0.8275
	MetaHIN	0.6019	0.7261	0.8893	0.7869	0.9593	0.8492	0.7915	0.9445	0.8385
	mp2vec	0.6669	0.8391	0.8144	0.8793	1.0968	0.8233	0.8972	1.1613	0.8235
	HERec	0.6518	0.8192	0.8233	0.8691	0.9916	0.8389	0.8894	1.0998	0.8265
	HAN	0.6537	0.8256	0.7921	0.9472	1.1402	0.7176	0.9438	1.1518	0.7500
	MvDGAE	**0.6009**	**0.7168**	**0.9059**	**0.7798**	**0.9526**	**0.8734**	**0.7814**	**0.9281**	**0.8635**
New items for existing users (Item Cold-start or IC)	FM	0.7186	0.9211	0.8342	1.3488	1.8503	0.7218	0.8293	1.1032	0.8122
	NeuMF	0.7063	0.8188	0.7396	0.9822	1.2042	0.6063	0.9273	1.1009	0.7722
	DropoutNet	0.7122	0.8021	0.8229	0.9604	1.1755	0.7547	0.8116	1.0301	0.7943
	GC-MC	0.9081	0.9702	0.7634	1.0433	1.2753	0.7062	0.8998	1.1043	0.8023
	MetaEmb	0.6741	0.7993	0.8537	0.9084	1.0874	0.8133	0.8055	0.9407	0.8092
	MeLU	0.6518	0.7738	0.8882	0.9196	1.0941	0.8041	0.7567	0.9169	0.8451
	MetaHIN	0.6252	0.7469	0.8902	0.8675	1.0462	0.8341	0.7174	0.8696	0.8551
	mp2vec	0.7371	0.9294	0.8231	1.0615	1.3004	0.6367	0.7979	1.0304	0.8337
	HERec	0.7481	0.9412	0.7827	0.9959	1.1782	0.7312	0.8107	1.0476	0.8291
	HAN	0.6619	0.8358	0.7787	0.9147	1.0857	0.7273	0.8126	1.0286	0.7574
	MvDGAE	**0.6122**	**0.7406**	**0.8947**	**0.8566**	**0.9789**	**0.8442**	**0.6952**	**0.8543**	**0.8827**
New items for new users (User-Item Cold-start or UIC)	FM	0.8326	0.9587	0.8201	1.3001	1.7351	0.7015	0.8363	1.1176	0.8278
	NeuMF	0.6949	0.8217	0.8566	0.9686	1.2832	0.8063	0.9860	1.1402	0.7836
	DropoutNet	0.8316	0.8489	0.8012	0.9635	1.1791	0.7617	0.8225	0.9736	0.8059
	GC-MC	0.7813	0.8908	0.8003	1.0295	1.2635	0.7302	0.8894	1.1109	0.7923
	MetaEmb	0.7733	0.9901	0.8541	0.9122	1.1088	0.8087	0.8285	0.9476	0.8188
	MeLU	0.6517	0.7752	0.8891	0.9091	1.0792	0.8106	0.7358	0.8921	0.8452
	MetaHIN	0.6318	0.7589	0.8934	0.8586	1.0286	0.8374	0.7195	0.8695	0.8521
	mp2vec	0.7987	1.0135	0.8527	1.0548	1.2895	0.6687	0.8381	1.0993	0.8137
	HERec	0.7859	0.9813	0.8545	0.9974	1.1012	0.7389	0.8274	0.9887	0.8034
	HAN	0.6588	0.8339	0.8003	0.9467	1.1404	0.6907	0.8320	1.0323	0.7559
	MvDGAE	**0.6201**	**0.7433**	**0.9054**	**0.8496**	**0.9869**	**0.8553**	**0.7074**	**0.8523**	**0.8722**
Existing items for existing users (Non-cold-start)	FM	0.7358	0.9763	0.8086	1.0043	1.1628	0.6493	0.8642	1.0655	0.7986
	NeuMF	0.6904	0.8373	0.7924	0.9249	1.1388	0.7335	0.7611	0.9731	0.8069
	DropoutNet	0.7108	0.7991	0.8268	0.9595	1.1731	0.7231	0.8219	1.0333	0.7394
	GC-MC	0.8056	0.9249	0.8032	0.9863	1.2238	0.7147	0.8518	1.0327	0.8023
	MetaEmb	0.7095	0.8218	0.7967	0.8086	1.0149	0.8077	0.7677	0.9789	0.7740
	MeLU	0.6519	0.7834	0.8697	0.8084	0.9978	0.8433	0.7382	0.9028	0.8356
	MetaHIN	0.6393	0.7704	0.8859	0.7997	0.9491	0.8499	0.6952	0.8445	0.8477
	mp2vec	0.6897	0.8471	0.8342	0.8788	1.1006	0.7091	0.7924	1.0191	0.8005
	HERec	0.6794	0.8409	0.8411	0.8652	1.0007	0.7182	0.7911	0.9897	0.8101
	HAN	0.6382	0.8249	0.7860	0.8968	1.0848	0.7377	0.7925	0.9943	0.7638
	MvDGAE	**0.5634**	**0.7069**	**0.9114**	**0.6987**	**0.8859**	**0.8801**	**0.6921**	**0.8339**	**0.8829**

Table 2: The AUCs of different methods on *WeChat Channel* Dataset. The larger AUC value means better performance. The best results of all methods are indicated in bold.

	GC-MC	mp2vec	HERec	HAN	MetaHIN	MvDGNN (w/o Reconstruction)	MvGAE (w/o Dropout)	MvDGAE$_n$ (w/o Bayesian)	MvDGAE
UC	0.5965	0.5973	0.6073	0.6136	0.6415	0.6189	0.6291	0.6458	**0.6563**
IC	0.6975	0.6891	0.6993	0.6958	0.7056	0.6959	0.7088	0.7186	**0.7219**
UIC	0.5956	0.6048	0.6067	0.6114	0.6443	0.6362	0.6453	0.6598	**0.6675**
Normal	0.6722	0.6917	0.7019	0.7046	0.7037	0.7103	0.7349	0.7307	**0.7374**

graph reconstruction objectives, i.e., only preserve the recommend loss $\mathcal{L}_{rec}$ in Eq. 11. (2) **MvGAE**: a variant of MvDGAE which remove the **dropout** operation on multi views, i.e., obtain the complementary representation by Eq. 3 instead of Eq. 4. (3) **MvDGAE**$_n$: a variant of MvDGAE which use a naive sum loss for multi-view reconstruction objectives, without the **Bayesian Task Weight Learner** to automatically balance. All the w_j and w_k are set to 1 in Eq. 7.

As shown in Fig 4 and Table 2, we observe that the full model MvDGAE achieves significant improvements across all metrics and datasets, which verify the effectiveness of each component. In detail, (1) Comparison of MvDGNN and MvDGAE: Without the **multi-view graph reconstruction objectives**, the performance degrades significantly most in both warm and cold-start scenarios. As we discuss above, the complementary representation further improves its ability with the help of multi-view graph reconstruction objectives. Thus each view semantic information describing the user/item relationships can be maintained. Moreover, the complementary representation is required to use the limited views to recover the missing one, which make it more robust. (2) Comparison of MvGAE and MvDGAE: In warm-start scenario, the performance of MvGAE is similar to MvDGAE. In cold-start scenario, we can observe that there are significant improvements with the help of the **Dropout** mechanism. We attribute it to the adaptation of the model on lacking user-item views during inference, i.e., the reducing on gaps between training and inference procedure. (3) Comparison of MvDGAE$_n$ and MvDGAE: In both warm and cold scenarios, the performance has been improved to a certain extent. It demonstrates the effectiveness of the Bayesian task weight learner in increasing the performance, in addition to its benefit that we do not need to manually set the weights for multi-view graph reconstruction objectives.

5.4 Case Study

Besides the performance effectiveness, MvDGAE has potentially good interpretability. Here we take two real examples in *WeChat Channel* to illustrate how the **multi views** mechanism work to alleviate the cold-start problem [5]. It helps us understand the recommendations provided by MvDGAE.

Example 1 (Recommend for New User). As shown in Fig. 5(a), the user (ID: 141926) is a new user and has not interacted with any videos. It is difficult to recommend meaningful and personalized videos for her. In MvDGAE, with the help of the **multi views** mechanism, we provide other auxiliary views to alleviate the user cold-start problem. Concretely, we leverage her social network and basic attributes (note that we would collect some basic attributes to form a combined feature) to find similar users. Then we will recommend the videos which her similar users are interested in to her.

Example 2 (Recommend for New Video). MvDGAE also can deal with the item cold-start problem greatly. As shown in Fig. 5(b), the new video (ID: 278533) has just been released and has not been interacted by any users. It is challenging to produce an informative embedding for it. However, MvDGAE leverages its auxiliary information, i.e., its publisher and tag, to find similar videos and thus obtain an informative and meaningful embedding. In this way, it is recommended to the users who are interested in similar videos.

[5]Note that the ids shown in the example are rearranged, instead of the original one.

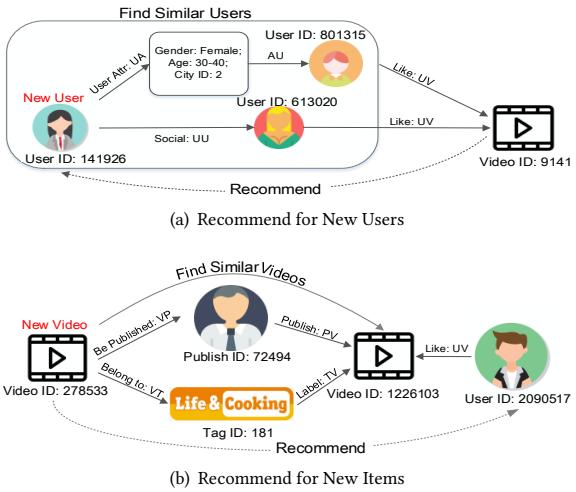

Figure 5: Case Study for New user and New item.

5.5 Parameter Analysis

In this section, we investigate the impact of different dropout rates and embedding dimensions on the recommendation performance. Here we use the large-scale industry datasets *WeChat Channel* as the case.

Impact of Dropout Rate. We plot the performance of MvDGAE under the setting of dropout rate from 0 to 1 in different scenarios, including warm and cold-start. In Fig 6 (a), we observe that in the warm-start scenario, the performance is generally stable when the dropout rate is in the range of 0 to 0.4. If the dropout rate continue to grow, the performance degrades rapidly. On the other hand, in the cold-start scenario, the performance improves with the growth of the dropout rate until 0.6. And then it remains stable despite the increase of dropout rate. The patterns reflect that our proposed model with dropout operation can maintain the accuracy in the warm-start but improve significantly in the cold-start scenarios.

Impact of Embedding Dimensions. We also explore how the dimensions of target nodes, i.e., user/item embeddings would affect the performance. As shown in Fig 6 (b), MvDGAE achieves optimal performance when the dimension is set to 128. Meanwhile, around

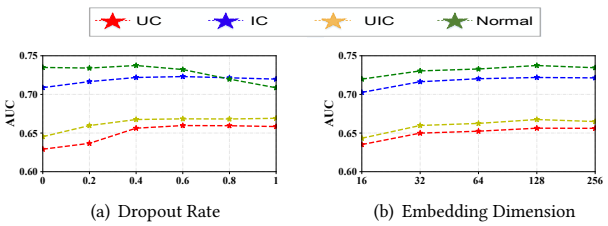

Figure 6: Analysis of parameters in MvDGAE.

the optimal setting, the performance is generally stable, which indicates that our model is robust to the embedding dimensions.

6 CONCLUSION

We propose a novel recommendation model named MvDGAE on HINS for the cold-start problem. The principal idea is to regard the cold-start as a missing data problem, i.e., some user-item interaction links are missing. After extracting the multi-views of both users and items from HINs, we randomly drop out some user-item interaction views when training the encoder while forcing the decoder to use the limited views to recover the full views. The decoder of MvDGAE is trained to reconstruct some user-user and item-item relationship graphs rather than the original input graphs, which make the features of the similar users (or items) in the meta-paths closer together. We finally adopt a Bayesian task weight learner to balance multi-view graph reconstruction objectives automatically. Extensive experiments on three public benchmark datasets and one large-scale industry dataset *WeChat Channel* verify the effectiveness and interpretability in various cold-start scenarios.

ACKNOWLEDGMENTS

The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant Nos. 61502174, 61872148), the Natural Science Foundation of Guangdong Province (Grant Nos. 2017A030313355, 2017A030313358, 2019A1515010768, 2021A1515011496), the Guangzhou Science and Technology Planning Project (Grant Nos. 201704030051, 201902010020), the Key R&D Program of Guangdong Province (Grant No. 2018B010107002), and the Fundamental Research Funds for the Central Universities. Jiawei Zheng is supported by 2020 Tencent Rhino-Bird Elite Training Program.

REFERENCES

[1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
[2] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. MAMO: Memory-Augmented Meta-Optimization for Cold-start Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 688–697.
[3] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.
[4] Shaohua Fan, Xiao Wang, Chuan Shi, Emiao Lu, Ken Lin, and Bai Wang. 2020. One2multi graph autoencoder for multi-view graph clustering. In *Proceedings of The Web Conference 2020*. 3070–3076.
[5] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2478–2486.
[6] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1797–1806.
[7] Palash Goyal and Emilio Ferrara. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* 151 (2018), 78–94.
[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
[9] Binbin Hu, Yuan Fang, and Chuan Shi. 2019. Adversarial learning on heterogeneous information networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 120–129.
[10] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 667–676.
[11] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7482–7491.
[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[13] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1073–1082.
[14] Cheng-Te Li, Chia-Tai Hsu, and Man-Kwan Shan. 2018. A cross-domain recommendation mechanism for cold-start users based on partial least squares regression. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 6 (2018), 1–26.
[15] Jorge M Lobo, Alberto Jiménez-Valverde, and Raimundo Real. 2008. AUC: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography* 17, 2 (2008), 145–151.
[16] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1563–1573.
[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013), 3111–3119.
[18] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 695–704.
[19] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 635–644.
[20] Amin Salehi and Hasan Davulcu. 2019. Graph attention auto-encoders. *arXiv preprint arXiv:1905.10715* (2019).
[21] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2018), 357–370.
[22] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2016. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2016), 17–37.
[23] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.
[24] Joaquin Vanschoren. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548* (2018).
[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[26] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* 11, 12 (2010).
[27] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. In *Advances in neural information processing systems*. 4957–4966.
[28] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 592–600.
[29] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.
[30] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. 2020. Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark. *IEEE Transactions on Knowledge and Data Engineering* (2020).
[31] Carl Yang, Jieyu Zhang, Haonan Wang, Sha Li, Myungwan Kim, Matt Walker, Yiou Xiao, and Jiawei Han. 2020. Relation learning on social networks with multi-modal graph edge variational autoencoders. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 699–707.
[32] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.
[33] Jun Zhao, Zhou Zhou, Ziyu Guan, Wei Zhao, Wei Ning, Guang Qiu, and Xiaofei He. 2019. Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2347–2357.

A APPENDIX

A.1 Pseudocode of MvDGAE

The pseudocode of the MvDGAE training procedure is described in Algorithm 1. As Algorithm 1 shows, we first sample the neighborhoods based on meta-path described in Section 4.2.1 in line 4. Based on meta-path, we leverage the node-level attention mechanism to obtain representation in line 5. Then we fuse all the path embedding with dropout operation to obtain the complementary representations in line 7. Finally, we leverage the representations $H^{(U)}$ and $H^{(I)}$ to finish the reconstruction objectives and recommend task.

Algorithm 1 MvDGAE Training Method

Input: An HIN $G = (V, E)$; meta-path set for users and items P_U, P_I, respectively.

1: Initialize all parameters.
2: **repeat**
3: **for** each user meta-path $p_k^{(u)} \in P_U$ **do**
4: Sample the neighborhoods based on meta-path $p_k^{(u)}$ according to Section 4.2.1.
5: Obtain the aggregated embedding $H_k^{(U)}$ based on the meta-path $p_k^{(u)}$.
6: **end for**
7: Fuse all path embeddings with the dropout operation to obtain the complementary representations $H^{(U)}$ according to Eq. 3.
8: $H^{(I)}$ are obtained similar to the $H^{(U)}$ based on the item meta-path set P_I.
9: Calculate loss $\mathcal{L}$ according to Eq. 11, and Back propagation.
10: **until** convergence

A.2 The Statistics of Datasets

Table 3: Statistics of the three public benchmark datasets and a large-scale industry dataset. The underlined node type refers to the target item type for recommendation.

Datasets	Node type	#Nodes	Edge type	#Edges	Density
DBook	User (U)	10,592	UB	649,381	2.9×10^{-3}
	Book (B)	20,934	BA	20,934	
	Author (A)	10,544	UU	169,150	
MovieLens	User (U)	6,040	UM	1,000,209	4.3×10^{-2}
	Movie (M)	3,881	MA	15,398	
	Actor (A)	8,030	MD	4,210	
	Director (D)	2,186			
Yelp	User (U)	51,624	UB	1,301,869	7.4×10^{-4}
	Business (B)	34,199	BC	34,199	
	City (C)	510	BT	103,150	
	Category (T)	541			
WeChat	User (U)	2,233,912	UU (Social)	45,221,042	8.6×10^{-6}
	Video (V)	1,854,117	UV	35,623,177	
	Publisher (P)	118,501	VP	1,854,117	
	Tag (T)	202	VT	4,022,156	

A.3 The details of Baselines

We compare MvDGAE with three categories of methods: **(1) Traditional methods**, including FM [19], NeuMF [8], DropoutNet [27], and GC-MC [1]. As they are not designed for HINs, following [16], we take the heterogeneous information as the features of users or items. **(2) Meta-learning based methods**, including MeteEmb [18], MeLU [13] and MetaHIN [16]. **(3) HIN-based methods**, including metapath2vec [3], HERec [21], HAN [29]. The details of these baselines are as follows:

- FM [19]: is a feature-based baseline which is able to utilize various kinds of auxiliary information. The rank of the factorization used for the second order interactions is set as 8 and utilize L2 regularization with coefficients 0.1.
- NeuMF [8]: consists of a generalized matrix factorization component and a MLP component. The layers are set to (64, 32, 16, 8) and learning rate 0.001.
- DropoutNet [27]: is a neural network based model for cold-start problem. The learning rate in DropoutNet is set to 0.9 and the dropout rate is set to 0.5.
- GC-MC [1]: a graph-based auto-encoder framework for matrix completion. The number of hidden units in the first and second layer are set to 500 and 75, respectively. The dropout fraction is set to 0.7.
- MeteEmb [18]: is a meta-learning based methods for CTR prediction. In MeteEmb, the coefficient α is set to 0.1.
- MeLU [13]: alleviate the user cold-start problem by adopting the MAML concept in the recommender system. Two layers for decision making layers are 64 nodes each, and the local update step is set as 1.
- MetaHIN [16]: a meta-learning approach to cold-start recommendation on HINs. The embedding dimension and meta-learning rate are set to 32 and 0.0005, respectively.
- metapath2vec [3]: applied meta-path based random walks for learning node embeddings.The length of random walk, the number of walks and the size of windows are set to 40, 10 and 3, respectively.
- HERec [21]: a novel heterogeneous network embedding based approach for HIN based recommendation. The tuning coefficients in HERec (i.e., α and β) are set to 1.0, and the random walk settings are same as in mp2vec.
- HAN [29]: introduced hierarchical attention to capture node-level and semantic-level information. The number of attention heads, learning rate, dropout rate are set to 8, 0.005, 0.6, respectively.

A.4 Implementation Details

Hyper-parameter Settings. We adopt Adaptive Moment Estimation (Adam) [12] to optimize our MvDGAE. For all dataset, we use a batch size of 512 and set the learning rate to 0.005. The embedding dimension of MvDGAE is set to 128 and the dropout rate is 0.4. Note that we also analyse the impact of hyper-parameters in MvDGAE in Section 5.5.

Environment Settings. We implement the proposed MvDGAE on Tensorflow [6] v1.15. and Python v3.6, respectively. And we use the large-scale graph learning framework named **PlatoDeep** to handle the graph operation effectively.

[6] https://www.tensorflow.org

Heterogeneous Auxiliary Relationships. As mentioned in Section 4, we leverage the meta-path extracted from the HINs to guide the GNN to obtain aggregated embedding. In addition, each meta-path is translated to the user-user or item-item graph to serve as the supervised reconstruction objectives. Here we provide the kinds of meta-path in Table 4 for different datasets. Note that we set the the number of common auxiliary neighbors for user-item-user and item-user-item to 3 to construct user-user and item-item graph, otherwise is 1. We collect some basic attributes of users and items to form a combined feature as the connection bridge.

Table 4: Kinds of heterogeneous auxiliary relationships

Dataset	User Aspect	Item Aspect
DBook	user-book-user user-attribute-user	book-user-book book-author-book
MovieLens	user-movie-user user-attribute-user	movie-user-movie movie-actor-movie movie-director-movie
Yelp	user-business-user user-attribute-user	business-user-business business-city-business business-category-business
WeChat	user-video-user user-user-video user-attribute-user	video-user-video video-publisher-video video-tag-video

A.5 The Detail Derivation of Bayesian Task Weight Learner

In Section 4.4, we aim to assign each view reconstruction objective a learnable weight to automatically achieve the balance among multi views.

Specially, we apply the same assumption in [11], $\frac{1}{\lambda^2}\left(\exp\left(\frac{x}{\lambda^2}\right)+1\right) \approx (\exp(x)+1)^{\frac{1}{\lambda^2}}$. Based on the assumption, we can get the following approximations for the sigmoid function $\sigma(\cdot)$:

$$\begin{aligned}
\sigma\left(\frac{x}{\lambda^2}\right) &= \frac{\exp\left(\frac{x}{\lambda^2}\right)}{\exp\left(\frac{x}{\lambda^2}\right)+1} \approx \frac{1}{\lambda^2}\left(\frac{\exp(x)}{\exp(x)+1}\right)^{\frac{1}{\lambda^2}} = \frac{1}{\lambda^2}(\sigma(x))^{\frac{1}{\lambda^2}} \\
1-\sigma\left(\frac{x}{\lambda^2}\right) &= \frac{1}{\exp\left(\frac{x}{\lambda^2}\right)+1} \approx \frac{1}{\lambda^2}\left(\frac{1}{\exp(x)+1}\right)^{\frac{1}{\lambda^2}} = \frac{1}{\lambda^2}(1-\sigma(x))^{\frac{1}{\lambda^2}}.
\end{aligned} \tag{12}$$

As mentioned above, we treat the reconstruction as the binary classification task, i.e., predict whether there is a link between two nodes. The binary classification likelihood can be defined as follows (for simplicity, here we omit the view index and type):

$$\begin{aligned}
\Pr(\boldsymbol{A} \mid \hat{\boldsymbol{A}}) &= \prod_{\langle i,j\rangle \in \boldsymbol{A}} \Pr\left(a(i,j) \mid \hat{a}(i,j)\right) \\
&= \prod_{\langle i,j\rangle \in \mathcal{A}^+} \sigma\left(\hat{a}(i,j)\right) \cdot \prod_{\langle i,j\rangle \in \mathcal{A}^-} \left(1-\sigma\left(\hat{a}(i,j)\right)\right),
\end{aligned} \tag{13}$$

where $\mathcal{A}^+$ denotes the set of positive sample, i.e. $a_{i,j}=1$ and $\mathcal{A}^-$ denotes the set of negative sample, i.e. $a_{i,j}=0$. Following [11], we introduce a scalar λ into Eq. 12 to get a scaled version:

$$\Pr(\boldsymbol{A} \mid \hat{\boldsymbol{A}}, \lambda) = \prod_{\langle i,j\rangle \in \mathcal{A}^+} \sigma\left(\frac{\hat{a}(i,j)}{\lambda^2}\right) \cdot \prod_{\langle i,j\rangle \in \mathcal{A}^-} \left(1-\sigma\left(\frac{\hat{a}(i,j)}{\lambda^2}\right)\right). \tag{14}$$

The input is scaled by λ^2 and then the log likelihood can be written as:

$$\begin{aligned}
&\log \Pr(\boldsymbol{A} \mid \hat{\boldsymbol{A}}, \lambda) \\
&= \sum_{\langle i,j\rangle \in \mathcal{A}^+} \log\left(\sigma\left(\frac{\hat{a}(i,j)}{\lambda^2}\right)\right) + \sum_{\langle i,j\rangle \in \mathcal{A}^-} \log\left(1-\sigma\left(\frac{\hat{a}(i,j)}{\lambda^2}\right)\right) \\
&\approx \sum_{\langle i,j\rangle \in \mathcal{A}^+} \left[-2\log\lambda + \frac{1}{\lambda^2}\log\left(\sigma\left(\hat{a}(i,j)\right)\right)\right] + \\
&\quad \sum_{\langle i,j\rangle \in \mathcal{A}^-} \left[-2\log\lambda + \frac{1}{\lambda^2}\log\left(1-\sigma\left(\hat{a}(i,j)\right)\right)\right] \\
&= -\frac{1}{\lambda^2}L_r - 2(|\mathcal{A}^+| + |\mathcal{A}^-|)\cdot\log\lambda.
\end{aligned} \tag{15}$$

Benefiting from the Bayesian Task Weight Learner, each view of user and item is assigned a learnable weight to automatically achieve the balance among multi views. Thus, the total reconstruction loss in Eq. 7 can be formulated as:

$$\begin{aligned}
\mathcal{L}_r &= \sum_j -\log\Pr(\boldsymbol{A} \mid \hat{\boldsymbol{A}}, w_j) + \sum_k -\log\Pr(\boldsymbol{A} \mid \hat{\boldsymbol{A}}, w_k) \\
&= \sum_j \left(\frac{1}{w_j^2}L_{rj}^{(u)} + 2\cdot\log w_j^2\right) + \sum_k \left(\frac{1}{w_k^2}L_{rk}^{(i)} + 2\cdot\log w_k^2\right),
\end{aligned} \tag{16}$$