MVGCL: Multi-View Graph Contrastive Learning for Service Recommendation

Guobing Zou[1], Shiyi Lin[1*], Shengxiang Hu[1], Yanglan Gan[2*], Bofeng Zhang[3], Yixin Chen[4]
[1]School of Computer Engineering and Science, Shanghai University, Shanghai, China
[2]School of Computer Science and Technology, Donghua University, Shanghai, China
[3]School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China
[4]Department of Computer Science and Engineering, Washington University in St. Louis, MO, USA
{gbzou, sylin, shengxianghu}@shu.edu.cn, ylgan@dhu.edu.cn, bfzhang@sspu.edu.cn, chen@cse.wustl.edu

***Abstract*—In service recommender system, graph neural networks (GNNs) perform message passing through diffusion mechanism based on user-service relationship graph. However, existing GNN-based service recommendation models suffer from two limitations: (1) message passing is only carried out at first-order neighbors, as higher-order may cause over-smoothing phenomenon, confining feature propagation in GNNs; and (2) due to sparse and noisy interactions, the distribution of embedding vectors is nonuniform in the latent space, resulting in unsatisfactory performance for downstream applications. To this end, we propose a fixed global graph diffusion view that is independent of the original user-service observed local view to form a multi-view learning by building contrastive learning (CL) relationship, named as Multi-View Graph Contrastive Learning (MVGCL). Specifically, it enhances the capability of message passing through constructed local and global multi-view graphs, and alleviates the sparse and noisy influences by performing intra-CL within local/global view and inter-CL between multi-view to obtain a more uniform distribution of user and service node representations. Extensive experiments are conducted on three benchmark datasets within different scales, and the results demonstrate that our proposed MVGCL can remarkably outperforms state-of-the-art competing baselines on various evaluation metrics.**

***Index Terms*—Service Recommendation, Multi-view Graph, Contrastive Learning, Collaborative Filtering, Graph Neural Network**

I. Introduction

With the development of smart service, "big data + AI algorithms" social infrastructure has been applied to service-oriented downstream tasks [1], such as service discovery, selection, composition, recommendation, and mashup creation [2]–[5]. Service recommendation techniques, which effectively alleviate the issues of information overload and users' unspecified needs, have been generally applied in the field of e-commerce and social media [6], [7].

Collaborative filtering (CF), as a fundamental approach, utilizes the prior knowledge of collective intelligence to recommend services for target users based on the preferences of groups with similar interests, which leverages recorded user-service interactions to predict unrecorded interactions, such as click-through rates, ratings, etc. Matrix factorization (MF) [8], from the linear mutual learning perspective of CF, firstly projects users and services into the latent space independently, and then connects their latent features by using dot product operation, achieving the purpose of modeling user-service interactions. Furthermore, with the considerable success made in deep neural networks (DNNs) [9], enormous efforts [10], [11] have been devoted to yielding collaborative signals by modeling complex nonlinear interactions. Thus, deeper semantic features of users and services can be extracted for diverse service recommendations, including QoS prediction and mashup service creation.

Graph collaborative filtering [12], [13], which exploits the user-service graph structure, focuses on node representations made by collaborative signal rather than modeling complex nonlinear interactions between users and services for better recommendation performance. Despite the recent advancements of graph collaborative filtering, it still suffers from two issues. First, data sparsity and noise cause distorted data distribution, resulting in unreliable node representation. Contrastive learning (CL), has shown usefulness to deal well with data sparsity in recommender systems. To this end, graph collaborative contrastive learning approaches [14] set up an auxiliary self-supervised task from graph structure. While partially alleviating the data sparsity problem, the new generated graphs may bring unreliable nodes or edges destroying the original graph structure. Second, the existing works on graph collaborative contrastive learning recommendation [14], [15] neglect the influence of high-order nodes. However, it has been shown to be essentially useful in recommendation tasks [16], [17], due to the over-smoothing problem [18], [19] caused by deep graph convolutional networks. So how to construct more effective graph collaborative contrastive learning for service recommendation has become a research challenge to be solved.

To address the above two issues, we explore the way of enriching first-order homogeneous neighbors of users/services based on multi-view graph contrastive learning. We propose a novel framework called MVGCL, which combines both local and global graph information with contrastive learning. Specifically, we design multi-view channels and CL targets, where LightGCN [13] is implemented as the backbone of graph channels. In multi-view channels, we initially utilize the original user-service interactions to form the local view,

*Corresponding author.

and then construct the global view through graph diffusion mechanism based on local view. In multi-level CL, we design dual objectives to catch the correlations between a node and its second-order neighbor ones (or first-order homogeneous ones) at the intra-view level, and to capture the relationships between local and global view at the inter-view level. The main contributions of this paper are summarized as follows.

- We propose a novel multi-view channel model for service recommendation to alleviate the over-smoothing problem, where high-order information can be fully fused. The local view channel focuses on low-order user/service neighbors, while the newly generated global view channel provisions high-order information based on graph diffusion.
- We design dual contrastive learning objectives to mitigate the influence of data sparsity and noise. One is between a node and its first-order homogeneous user/service neighbors within local or global views at the intra-view level, and the other is between a node in the local view and its corresponding node in the global view at the inter-view level.
- Extensive experiments are conducted on three benchmark datasets, demonstrating that our proposed approach receives superior performance for service recommendation tasks over multiple state-of-the-art approaches.

The remainder of this paper is organized as follows. Section II formulates the problem and preliminary. Section III illustrates the overall framework of MVGCL and presents the methodology. Section IV shows and analyzes the experimental results. Section V reviews the related work. Finally, Section VI concludes the paper and discusses the future work.

II. Problem Formulation and Preliminary

In this section, we firstly formulates problem definition of graph link prediction for downstream service recommendation task, and then provides the preliminary of graph contrastive learning (GCL).

A. Graph Link Prediction

We denote the user-service interaction matrix as $\mathbf{R} \in \mathbb{R}^{N \times M}$, where N and M are the number of users and services, respectively. Here, $\mathbf{R}$ in implicit feedback can be described as:

$$r_{u,s} = \begin{cases} 1 & \text{if observed;} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where 1 indicates there is an interaction between user u and service s; 0 indicates user u has no interaction with service s. The goal is to seek a model that predicts a list $\mathbf{y} = [y_1, ..., y_k, ..., y_m]$ for each user. $y_k \in \mathbb{R}^1$ is a score. We choose the top-K services from $\mathbf{y}$ for its corresponding user.

Based on the above user-service interaction matrix, let $\mathcal{G} = (U, V, E)$ be an undirected bipartite network, where U and V denote the set of users and services, and an edge e in E denotes an observed interaction between a user u and a service v. To facilitate the operation of user-service interaction matrix [12], the symmetric adjacency matrix of the user-service graph is represented as:

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R^T} & \mathbf{0} \end{pmatrix} \tag{2}$$

In such case, service recommendation task by graph link prediction is to identify unobserved top-K links for each u belonging to U, given observed link set E, user set U and service set V.

B. Graph Contrastive Learning

The core of contrastive learning (CL) is to attract positive pairs and pull away negative pairs. Graph Contrastive Learning (GCL) tries to bring the same sample from different views close to each other in the latent representation space and maximize the distance between dissimilar samples. A typical contrastive loss InfoNCE [20]:

$$\mathcal{L}_{CL} = -\mathbb{E}_{\mathtt{x}}[log \frac{\exp(d(x_i, x_i')/\tau)}{\exp(d(x_i, x_i')/\tau) + \sum_{j \neq i} \exp(d(x_i, x_j')/\tau)}] \tag{3}$$

where $d(\bullet)$ denotes distance metric function, such as Euclidean distance, and τ is the temperature parameter that adjusts the attention level to hard samples.

III. Methodology

Fig. 1 illustrates the overall framework of MVGCL. It consists of two crucial components: multi-view feature propagation and layers & views combination. More specifically, multi-view feature propagation is designed for incorporating low and high order interaction information, including local graph message passing, global graph construction and message passing. Here, intra-CL is considered in the same view to attract the embedding of first-order homogeneous neighbors. Based on the layer embedding from multi-view feature propagation, layers & views combination is conducted by combining various layers and views embedding with inter-CL, which maximizes the agreement between local and global views of the same node. To achieve the goal of service recommendation, we adopt multi-task training to make predictions by integrating the losses of supervised BPR and self-supervised CL.

A. Local Graph Message Passing

We initially represent a user u and a service s with a randomized feature vector $e_u^{(0)} \in \mathbb{R}^d$ and $e_s^{(0)} \in \mathbb{R}^d$, respectively, where d indicates the dimension of feature vector. Considering that users' direct interests are captured in the observed user-service interactions, we adopt the observed local graph and utilize GNN to model the interactive relationships between users and services in MVGCL. In GNN feature propagation, following LightGCN [13], we discard the nonlinear activation and feature transformation in the message passing function as:

$$\begin{aligned} \dot{e}_u^{(k+1)} &= \sum_{s \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_s|}} \dot{e}_s^{(k)} \\ \dot{e}_s^{(k+1)} &= \sum_{u \in N_s} \frac{1}{\sqrt{|N_s|}\sqrt{|N_u|}} \dot{e}_u^{(k)} \end{aligned} \tag{4}$$

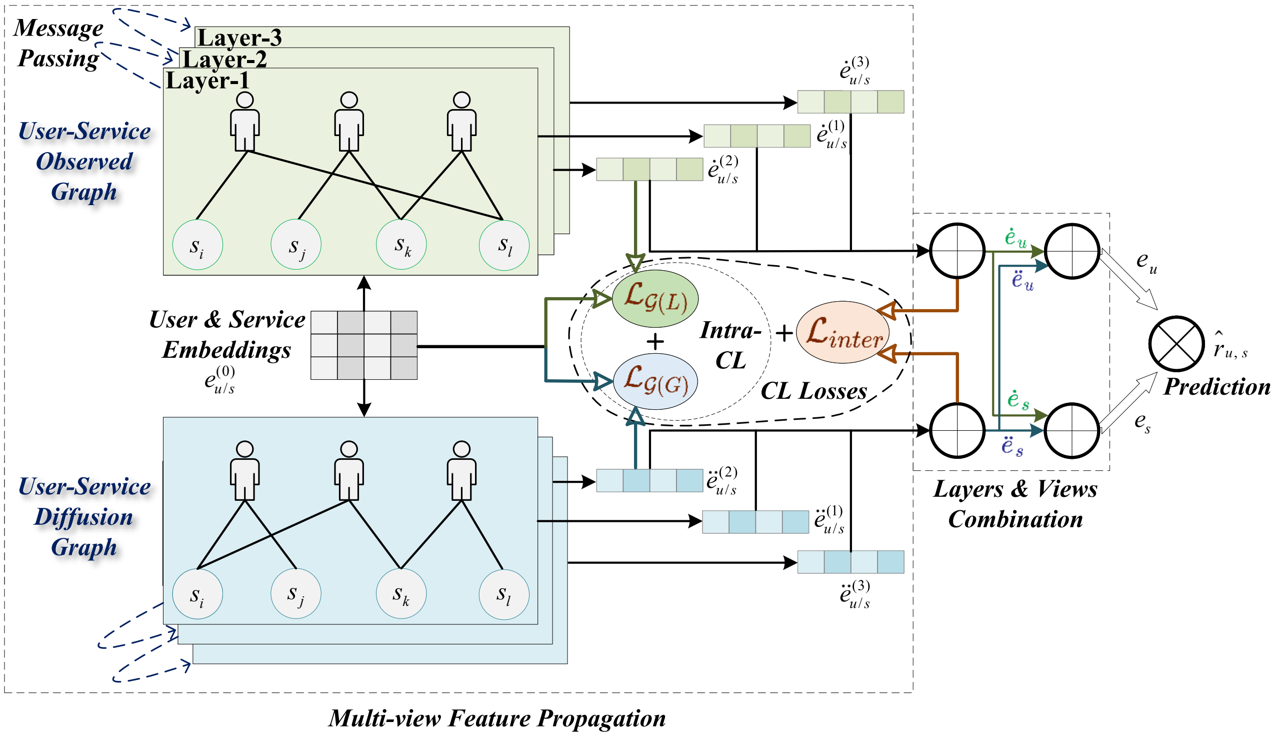


Fig. 1: Overall framework of multi-view graph contrastive learning for service recommendation.

where $|N_u|$ and $|N_s|$ denote the number of adjacent vertices that are directly connected to u and s in user-service observed local graph. Here, $e_u^{(0)}$ and $e_s^{(0)}$ are fed as $\dot{e}_u^{(0)}$ and $\dot{e}_s^{(0)}$ in initial layer-0, and multi-layer embeddings of users and services can be obtained by Equation (4).

In user-service observed local graph, there is the most correlation between a target user or service with the first-order homogeneous neighbor nodes. In view of the common interests that exist in the above nodes, we perform CL learning function between layer-0 and layer-2 representations of users and services. The local user CL loss is as follows:

$$\mathcal{L}^{U}_{\mathcal{G}(L)} = \sum_{u \in \mathcal{U}} -log \frac{\exp((\dot{e}_u^{(0)} \cdot \dot{e}_u^{(2)})/\tau)}{\sum_{v \in \mathcal{U}} \exp((\dot{e}_v^{(0)} \cdot \dot{e}_u^{(2)})/\tau)} \tag{5}$$

where $\dot{e}_u^{(0)}$ and $\dot{e}_v^{(0)}$ are normalized GNN layer-0 representations; $\dot{e}_u^{(2)}$ is normalized GNN layer-2 representation; τ is the temperature parameter. In the same way, the local service CL loss is expressed as:

$$\mathcal{L}^{S}_{\mathcal{G}(L)} = \sum_{s \in \mathcal{S}} -log \frac{\exp((\dot{e}_s^{(0)} \cdot \dot{e}_s^{(2)})/\tau)}{\sum_{t \in \mathcal{S}} \exp((\dot{e}_t^{(0)} \cdot \dot{e}_s^{(2)})/\tau)} \tag{6}$$

By integrating the above two losses from users and services, the local CL loss is summed together as below:

$$\mathcal{L}_{\mathcal{G}(L)} = \mathcal{L}^{U}_{\mathcal{G}(L)} + \mathcal{L}^{S}_{\mathcal{G}(L)} \tag{7}$$

B. Global Graph Construction and Message Passing

Given an observed user-service interaction A, message passing function can be expressed in Laplacian matrix form:

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \tag{8}$$

where $\mathbf{A}$ is the observed user-service interaction, $\mathbf{D}$ is the diagonal matrix with $D_{ii} = \sum_j A_{ij}$. Based on generalized Laplacian matrix, the diffusion matrix can be generated and expressed:

$$\mathbf{G} = \sum_{k=0}^{\infty} \theta_k \mathbf{L}^k \tag{9}$$

where θ_k is the weighting coefficient for each layer in the graph.

We use Personlizsed PageRank (PPR) as decay strategy: $\theta_k = \alpha(1-\alpha)^k$. By approximating the diffusion matrix $\mathbf{G}$ by Equation (9) based on the PPR [16], we can transmit it as a numerical diffusion matrix:

$$\mathbf{G}^* = \alpha(\mathbf{I_n} - (1-\alpha)\mathbf{L})^{-1} \tag{10}$$

where α is the decay factor through each layer message passing, and $\mathbf{I_n}$ is a diagonal matrix.

In numerical diffusion matrix $\mathbf{G}^*$, a smaller value in row u and column s indicates a lower possibility that u interacts with s. We design two kinds of user-based and service-based global interaction diffusion graphs that apply the above numerical diffusion matrix $\mathbf{G}^*$ to observed user-service interactive local graph.

- **MVGCL-U ($\mathbf{G_U}$):** For each row u in U, we firstly obtain the degree $|N_u|$ of u from observed local graph, and then retain top-$|N_u|$ service nodes as one-hop neighbor set. By this way, user-based approach identifies the most relevant services based on observed user-service locally direct interactions. It can be formulated as:

$$\mathbf{G_U}_{u,:} = \mathbb{I}(argmax(\mathbf{G}^*, deg(\mathbf{A}_{u,:}))) \tag{11}$$

where $\mathbb{I}(x)$ is an indicator function: $\mathbb{I}(x) = 1$ if x is true, and 0 otherwise. And $deg(\mathbf{A}_{u,:})$ is the degree of user u in adjacency matrix $\mathbf{A}$.

- **MVGCL-S ($\mathbf{G_S}$):** For each column s in S, we firstly obtain the degree $|N_s|$ of s from observed local graph, and then retain top-$|N_s|$ user nodes as one-hop neighbor set. By this way, service-based approach can find the most relevant users based on observed user-service locally direct interactions. It can be expressed as:

$$\mathbf{G_S}_{:,s} = \mathbb{I}(argmax(\mathbf{G}^*, deg(\mathbf{A}_{:,s}))) \tag{12}$$

where $deg(\mathbf{A}_{:,s})$ is the degree of service s in $\mathbf{A}$.

To reveal the potential exposure of user and service nodes, we set a graph diffusion convolution plus parameter k^* to enlarge the global neighborhood by proportion. Here, the expansion degree for each user u in $\mathbf{G_U}$ is expressed as:

$$V_u = |N_u| \times (1.0 + k^*) \tag{13}$$

Then, we use V_u as the degree of u to replace the $|N_u|$ in modeling user-oriented global diffusion graph $\mathbf{G_U}$. Similarly, it can also be applied in modeling service-oriented global diffusion graph $\mathbf{G_S}$.

After user-service global diffusion graph is constructed, global GNN is leveraged to perform message passing as:

$$\begin{aligned}\ddot{e}_u^{(k+1)} &= \sum_{s \in M_u} \frac{1}{\sqrt{|M_u|}\sqrt{|M_s|}} \ddot{e}_s^{(k)} \\ \ddot{e}_s^{(k+1)} &= \sum_{u \in M_s} \frac{1}{\sqrt{|M_s|}\sqrt{|M_u|}} \ddot{e}_u^{(k)}\end{aligned} \tag{14}$$

where $e_u^{(0)}$ and $e_s^{(0)}$ are used to represent the initialized vectors of global graph as $\ddot{e}_u^{(0)}$ and $\ddot{e}_s^{(0)}$; $|M_u|$ and $|M_s|$ denote the number of adjacent vertices that are directly connected to u and s in user-service diffusion graph $\mathbf{G_{U/S}}$.

Similarly, following Equation (5) and (6) by global embeddings of users and services, we can obtain $\mathcal{L}_{\mathcal{G}(G)}$ by combining $\mathcal{L}^U_{\mathcal{G}(G)}$ and $\mathcal{L}^S_{\mathcal{G}(G)}$:

$$\mathcal{L}_{\mathcal{G}(G)} = \mathcal{L}^U_{\mathcal{G}(G)} + \mathcal{L}^S_{\mathcal{G}(G)} \tag{15}$$

C. Multiple Layers & Views Combination

After propagating with multiple layers, we carry out the combination of all layers representations by adopting the weighted sum function for GNN readout in both local and global views:

$$\dot{e}_u = \frac{\sum_{h=0}^{H} \dot{e}_u^{(h)}}{H+1},\ \dot{e}_s = \frac{\sum_{h=0}^{H} \dot{e}_s^{(h)}}{H+1} \tag{16}$$

$$\ddot{e}_u = \frac{\sum_{h=0}^{H} \ddot{e}_u^{(h)}}{H+1},\ \ddot{e}_s = \frac{\sum_{h=0}^{H} \ddot{e}_s^{(h)}}{H+1} \tag{17}$$

Here, H is the number of layers and empirically fixed to three according to LightGCN [13]. The purpose is to capture a user's actual intent by multi-view way, which undermines the noise effect from singleton view. Although the observed user-service interactive relationships are close to the actual data distribution, small amounts of noisy data still exist in the observed interactions. Therefore, it possibly leads to the phenomenon that the readout representations of the same node from local and global views are analogous, which can be enhanced by inter-CL:

$$\begin{aligned}\mathcal{L}^U_{inter} = &\sum_{u \in \mathcal{U}} -log \frac{\exp((\dot{e}_u \cdot \ddot{e}_u)/\tau)}{\sum_{v \in \mathcal{U}} \exp((\dot{e}_v \cdot \ddot{e}_u)/\tau)} \\ &+ \sum_{u \in \mathcal{U}} -log \frac{\exp((\ddot{e}_u \cdot \dot{e}_u)/\tau)}{\sum_{v \in \mathcal{U}} \exp((\ddot{e}_v \cdot \dot{e}_u)/\tau)}\end{aligned} \tag{18}$$

$$\begin{aligned}\mathcal{L}^S_{inter} = &\sum_{s \in \mathcal{S}} -log \frac{\exp((\dot{e}_s \cdot \ddot{e}_s)/\tau)}{\sum_{v \in \mathcal{S}} \exp((\dot{e}_t \cdot \ddot{e}_s)/\tau)} \\ &+ \sum_{s \in \mathcal{S}} -log \frac{\exp((\ddot{e}_s \cdot \dot{e}_s)/\tau)}{\sum_{v \in \mathcal{S}} \exp((\ddot{e}_t \cdot \dot{e}_s)/\tau)}\end{aligned} \tag{19}$$

$$\mathcal{L}_{inter} = \mathcal{L}^U_{inter} + \mathcal{L}^S_{inter} \tag{20}$$

Subsequently, we combine readout representations of local and global views to obtain the final representations of a user and service:

$$e_u = \frac{\dot{e}_u + \ddot{e}_u}{2},\ e_s = \frac{\dot{e}_s + \ddot{e}_s}{2} \tag{21}$$

With the final representations, the inner product operation is performed to predict the value of how likely a user may interact with a service:

$$\hat{r}_{u,s} = e_u^T \cdot e_s \tag{22}$$

D. Multi-task Training

Bayesian Pairwise Ranking (BPR) loss is used as the primary loss function, which takes information directly from the interaction. Specifically, for each observed user-service pair (u, s^+), we randomly sample a service s^- that has no interaction with u to form a triplet (u, s^+, s^-):

$$\mathcal{L}_{BPR} = \sum_{(u,s^+,s^-) \in O} -log\ \sigma(\hat{r}_{u,s^+} - \hat{r}_{u,s^-}) \tag{23}$$

where $O = \{(u, s^+, s^-)|(u, s^+) \in O^+, (u, s^-) \in O^-\}$ is the training data and σ is the sigmoid function.

We utilize a multi-task training strategy to jointly optimize the parameters by adding self-supervised to further auxiliarily perform BPR primary loss training:

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{\mathcal{G}(L)} + \lambda_2 \mathcal{L}_{\mathcal{G}(G)} + \lambda_3 \mathcal{L}_{inter} + \lambda_4 ||\theta||_2 \tag{24}$$

where λ_1, λ_2 are the hyper-parameters to control the local and global intra-CL, and λ_3 controls the strengths of inter-CL. λ_4 controls the L2 regularization, and $||\theta||$ denotes the parameters of GNN model.

IV. EXPERIMENTS

A. Expermental Settings

Experimental Datasets. To verify the performance of MVGCL, we conduct extensive experiments on three public datasets: MovieLens-100K (ML-100K)[1], MovieLens-1M (ML-1M)[2] and Yelp[3]. The ratings of the above three benchmarking datasets are on an integer scale of 1 to 5. In each dataset, we reset ratings less than 4 to 0, and otherwise 1. For Yelp dataset, we filter users and services with fewer than 20 interactions to ensure the data quality in the experiments. The statistics of the datasets are shown in Table I. We split the datasets into training set, validation set and test set with a ratio of 8:1:1.

Competing Methods. We compare our proposed MVGCL with the following seven state-of-the-art competing baselines, including three MF-based, two GNN-based and two GCL-based methods. They are described as:

- **MF-based CF methods:**
  - **BPRMF [8]:** It learns the latent representation with matrix factorization model that is optimized by Bayesian personalized ranking.
  - **NeuMF [10]:** It models complex nonlinear interactions for user-service interactive relationships by deep neural network (DNN).
  - **ENMF [11]:** It trains neural recommendation models from the overall training data without negative sampling.
- **GNN-based CF methods:**
  - **NGCF [12]:** It integrates user-service interactions into the embedding process, and leverages GCN as the feature propagation function.
  - **LightGCN [13]:** It is the simplified version of the NGCF, which is more concise and suitable for service recommendation.
- **GCL-based CF methods:**
  - **SGL [14]:** It performs contrastive learning based on multiple augmentation graphs, which include node dropout, edge dropout and random walk.
  - **GDCL [21]:** It performs contrastive learning based on diffusion matrix, whose identical structure is considered in all nodes.

To prevent the deviations, we run competing baselines three times to calculate the average results for the guarantee of fair comparisons in the experiments.

Implementation Details. All the experiments are carried on our workstation equipped with two NVIDIA GTX 1080Ti GPU, an Intel(R) Xeon(R) Gold 6132 CPU@2.60 GHz and 192GB RAM. The components of MVGCL are implemented by Pytorch 1.7.1 with RecBole [22].

[1]https://grouplens.org/datasets/movielens/100k/
[2]https://grouplens.org/datasets/movielens/1m/
[3]https://www.yelp.com/dataset

TABLE I: Statistics of the experimental datasets.

Datasets	Users	Services	Interactions	Density
ML-100K	943	1,448	55,375	4.06%
ML-1M	6,039	3,934	575,281	2.42%
Yelp	17,523	13,114	765,807	0.33%

TABLE II: Parameter settings

Parameter	Value
τ	0.05 for ML-100K, 0.04 for ML-1M, 0.03 for Yelp
λ_1	1e-06 for ML-100K, 1e-06 for ML-1M, 1e-06 for Yelp
λ_2	1e-08 for ML-100K, 1e-08 for ML-1M, 1e-06 for Yelp
λ_3	1e-06 for ML-100K, 1e-06 for ML-1M, 1e-05 for Yelp
k^*	0.04 for ML-100K, 0.02 for ML-1M, 0.00 for Yelp

For a fair comparison, we refer to the best hyper-parameter settings reported in the original papers of the baselines and then fine-tune all the hyper-parameters of the baselines. As the general settings of all the baselines, the Xavier initialization [23] is used on all the embeddings. The batch size is 2048. The embedding size d is 64 and the model regularization parameter λ_4 is 1e-4. We use Adam with the learning rate 0.001 to optimize all the models. We tune the hyper-parameter temperature τ in $\{0.03, 0.04, 0.05, 0.075, 0.1\}$, SSL regularization λ_1, λ_2, λ_3 in {1e-5, 1e-6, 1e-7, 1e-8}, and k^* in $\{0.00, 0.02, 0.04, 0.06, 0.08, 0.10\}$. As for MVGCL, the parameters settings are shown in Table II.

Evaluation Metrics. Given a user u in U, $\hat{R}(u)$ indicates a ranked list of top-*K* services predicted by our model. $R(u)$ represents a ground-truth set of services that u has interacted with. We adopt widely-used top-*K* evaluations metrics, including Recall@*K* and NDCG@*K* (Normalized Discounted Cumulative Gain), where *K* is set to 20. Recall is defined as follows:

$$Recall@K = \frac{1}{|U|} \sum_{u \in U} \frac{|\hat{R}(u) \cap R(u)|}{|R(u)|} \quad (25)$$

Recall evaluates whether the predicted services appear in the set $R(u)$, which does not reflect the ranking quality. NDCG takes into account the position of the predicted services by assigning higher scores to the top-ranked hits:

$$NDCG@K = \frac{1}{|U|} \sum_{u \in U} N(u)_K \sum_{i=1}^{K} \frac{2^{\mathbb{I}(\hat{R}_i(u) \in R(u))} - 1}{log(i+1)} \quad (26)$$

where $N(u)_K = \sum_{i=1}^{min(K,|R(u)|)} \frac{1}{log(i+1)}$ is a normalizer to ensure that the perfect ranking has a value of 1; $\mathbb{I}(x)$ is an indicator function: $\mathbb{I}(x) = 1$ if x is true, and 0 otherwise.

B. Experiment Results and Analyses

To validate the effectiveness of MVGCL, we conduct extensive experiments on three datasets with different sizes and densities. Table III shows the comparison results of service recommendation among MVGCL and seven competing

TABLE III: Comparison results among MVGCL and seven competing baselines.

	ML-100K		ML-1M		Yelp	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
BPRMF	0.3334	0.2207	0.2970	0.1716	0.1172	0.0718
NeuMF	0.3365	0.2169	0.2494	0.1926	0.0890	0.0516
ENMF	0.3531	0.2367	0.2970	<u>0.2355</u>	0.1226	0.0760
NGCF	0.3388	0.2212	0.2781	0.2163	0.1129	0.0673
LightGCN	0.3097	0.1929	0.2939	0.2288	0.1309	0.0791
SGL	0.3546	<u>0.2371</u>	<u>0.2981</u>	0.2342	<u>0.1423</u>	0.0914
GDCL	<u>0.3552</u>	0.2312	0.2976	0.2326	0.1418	<u>0.0917</u>
MVGCL	**0.3729**	**0.2434**	**0.3100**	**0.2436**	**0.1494**	**0.0956**

TABLE IV: Results of ablation experiments among MVGCL and its four different variants.

	ML-100K		ML-1M		Yelp	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
L w/o $\mathcal{L}_{CL}$	0.3097	0.1929	0.2939	0.2288	0.1309	0.0791
G w/o $\mathcal{L}_{CL}$	0.3330	0.2228	0.2937	0.2290	0.1336	0.0821
L & G w/o $\mathcal{L}_{intra}$	0.3572	0.2322	0.2976	0.2336	0.1438	0.0925
L & G w/o $\mathcal{L}_{inter}$	0.3633	0.2373	0.3031	0.2397	0.1480	0.0952
MVGCL	**0.3729**	**0.2434**	**0.3100**	**0.2436**	**0.1494**	**0.0956**

baselines. Here, higher Recall and NDCG indicate better performance of top-K service recommendation. The best and second-best results of each column are marked in dark and underline, respectively. It is observed that recommendation metrics achieve better performance on the whole with the increasing density of experimental datasets.

MF-based competing baselines perform poorly on both Recall and NDCG, as it depends primarily on direct user-service interactions. Specifically, NeuMF models user-service interaction through nonlinear and multiple neural networks directly while representing features based on a single collaborative signal layer, which results in an overfitting model. Conversely, ENMF achieves better performance by inheriting the simplified BPRMF architecture and further optimizing the negative sample strategy. Although complex combinations are adopted downstream in this kind of baselines, encoding shallow relationships significantly constraints model effectiveness for service recommendation.

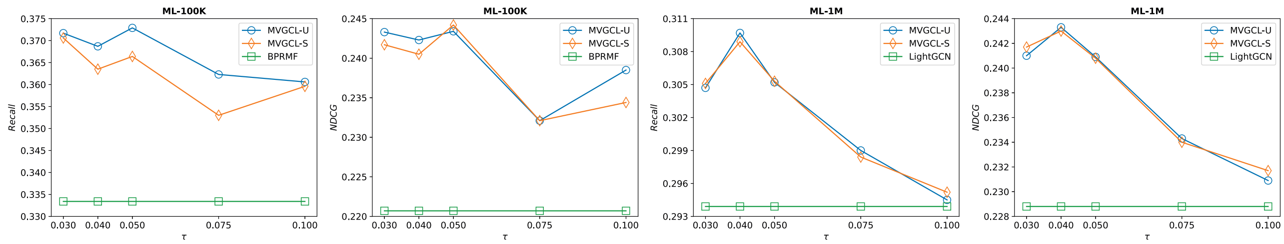
Compared to MF-based competing baselines, GNN-based methods have advantage that they encode high-order information is encoded into feature representations. Exceptionally, MF-based models are superior to GNN-based methods on ML-100K dataset, since it is sufficient to encode and integrate information from user-service direct collaborative interactions on a relatively dense dataset. Among all GNN-based methods, LightGCN achieves the best performance in most cases with the fastest efficiency of model training. The reason for the improvement in LightGCN is the elimination of feature transformation and nonlinear activation modules, which have a negative impact in the NGCF model. However, the performance of this kind of competing baselines relies heavily on the density of user-service interactions in experimental datasets. Thus, when applying in the scenarios with sparse interactions, they may produce non-uniform representations of users and services.

By performing contrastive learning in GNN, it is observed that SGL and GDCL consistently outperform other supervised competing baselines on three datasets. It indicates that contrastive learning plays an important role and brings the obvious improvement in top-K service recommendation. Nevertheless, SGL generates views by random node/edge dropouts and GDCL sets fixed degree for each node in user-service diffusion matrix, which extremely destroys the structure of user-service bipartite graph. Compared to GCL-based competing baselines, MVGCL performs intra-CL within local and global views and inter-CL between dual views, which can more effectively reflect the direct and indirect user-service interactive relationships. It is beneficial to message passing and feature propagation for better prediction accuracy.

Consequently, our proposed MVGCL receives the best performance among all competing baselines on Recall and NGCG across multiple datasets by a large margin, which demonstrates the effectiveness of considering multi-level contrastive learning in top-K service recommendation tasks.

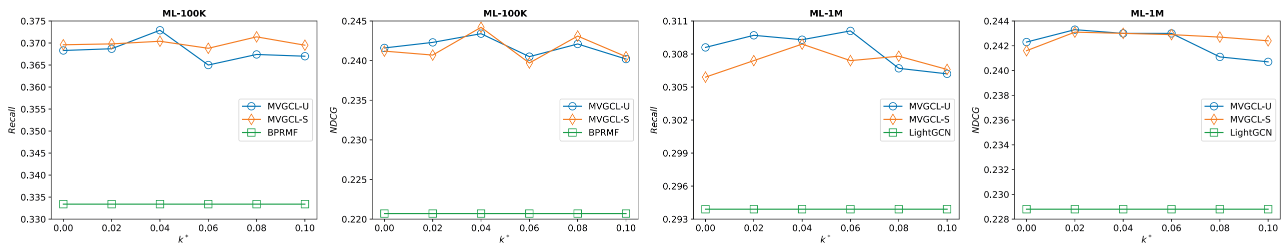
C. Ablation Study

To validate the effectiveness of MVGCL, ablation studies are conducted to analyze the performance impact of each component in MVGCL. Table IV reports the results of ablation experiments among MVGCL and its four different variants. In the experiments, "L w/o $\mathcal{L}_{CL}$" (LightGCN) represents the single local view backbone of MVGCL without CL loss; "G w/o $\mathcal{L}_{CL}$" indicates that message passing is performed only via a single global graph to replace local graph without CL loss; "L & G w/o $\mathcal{L}_{intra}$" and "L & G w/o $\mathcal{L}_{inter}$" are multi-



(a) Recall of ML-100K (b) NDCG of ML-100K (c) Recall of ML-1M (d) NDCG of ML-1M

Fig. 2: Performance impact of different τ on MVGCL under ML-100K and ML-1M.



(a) Recall of ML-100K (b) NDCG of ML-100K (c) Recall of ML-1M (d) NDCG of ML-1M

Fig. 3: Performance impact of different k^* on MVGCL under ML-100K and ML-1M.

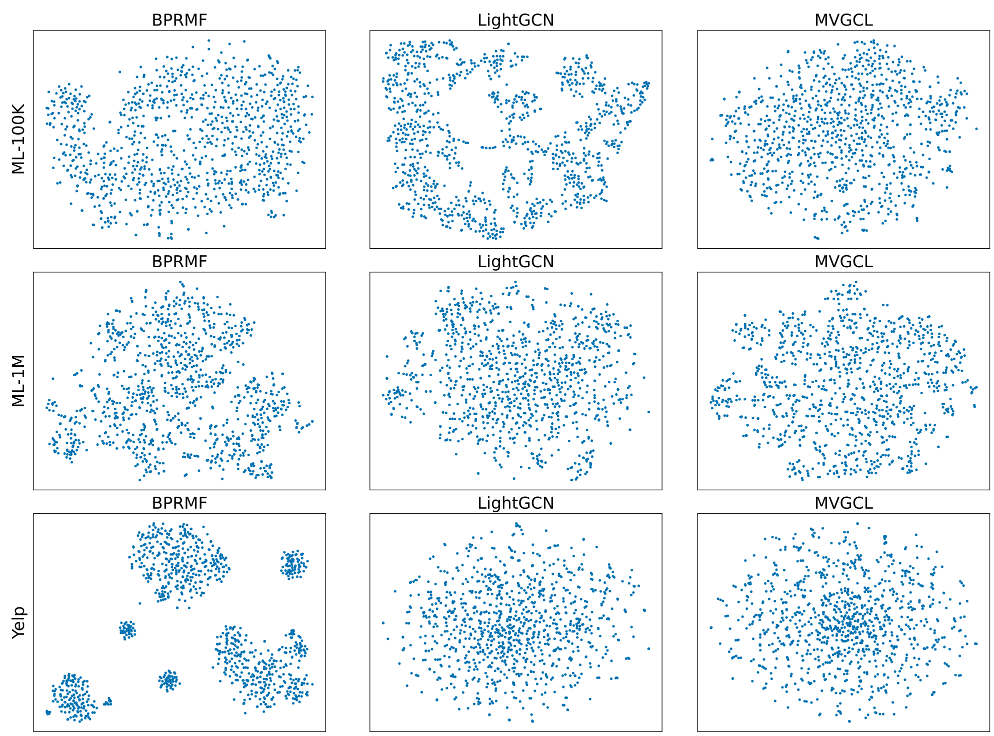


Fig. 4: Embedding distributions on ML-100K, ML-1M and Yelp visualized with t-SNE.

view propagation variants by removing intra-CL and inter-CL, respectively. We conclude from the experimental results that are twofold as below.

- A competitive recommendation performance can be maintained or even improved by replacing the locally observed adjacency matrix with the globally graph diffusion matrix. It shows that user-service graph diffusion matrix has the ability to eliminate noise and generate a realistic data distribution consistent with the observed data distribution.
- With the consideration of inter-CL or intra-CL, a better performance can be obtained than baseline LightGCN. Especially, intra-CL provides more significant performance

increment of service recommendation. It shows that a more uniform representation can be obtained than relying solely on label data in training phases, when applying contrastive learning to feature representation of users and services.

D. Performance Impact of Hyper-Parameters

1) Impact of Temperature: τ plays a key role in hard negative sample mining in contrastive learning. To analyze the performance impact of τ on MVGCL, we vary τ in $\{0.03, 0.04, 0.05, 0.075, 0.1\}$, and Fig. 2 shows the fluctuations on recommendation performance along with the variations of τ. It can be seen that a larger value of τ causes poor performance. The primary reason is that larger temperature values have difficulty in reasonably identifying hard and easy negative samples. From the above experiments, τ in [0.03, 0.05] leads to a good recommendation performance.

2) Impact of Graph Diffusion Convolution Plus: To test the performance impact of graph diffusion convolution plus k^*, we set it to the scope of $\{0.00, 0.02, 0.04, 0.06, 0.08, 0.10\}$ in the experiments. The fluctuations on recommendation performance along with the variations of k^* is shown in Fig. 3. It can be observed that both Recall and NDCG arise as k^* increases initially, and then decline with the increasing k^*. This phenomenon can be explained that sparse user-service interactions still undermine performance with overfitting training, when k^* is set too small. In another extreme case, when k^* is set to be a large value, abundant noisy user-service interactions possibly hurt the model learning since those newly generated interactions are not constantly reliable. Based on the above analyses, when k^* in MVGCL is set in the range of [0.02, 0.08] in our experiments, it achieves the best service recommendation performance.

E. Visualizing the Embedding Distributions

To better understand the benefits of MVGCL, we randomly sample the BPRMF, LightGCN, MVGCL embeddings of 1,000 service nodes from ML-100K, ML-1M, and Yelp datasets, and then project them into 2-D space with t-SNE [24]. As can be seen from Fig. 4, the embedding vectors learned by the BPRMF in low density datasets easily fall into several isolated clusters, and conversely, LightGCN fails to obtain uniform representation vectors at high densities. As shown in previous literature [25], there is a strong relationship between contrastive learning and uniformity of representations. Since non-CL methods cannot receive self-supervised signals, they have difficulty in balancing collaborative relationship and latent representation uniformity among service nodes, resulting in representation vectors easily falling into local optimum. Notably, MVGCL achieves uniform representation and optimally identifies the community structures as well in the latent representation space, boosting the service recommendation performance.

V. Related Work

In this section, we review the existing graph-based and graph contrastive learning (GCL-based) methods that are related to our MVGCL, respectively.

A. Graph-based Methods

Studies on graph-based methods can be classified into two categories: graph embedding and graph neural networks (GNNs). Graph embedding methods aim to learn the embedding features of nodes. DeepWalk [26] performs truncated random walks on the network to obtain a sequence of sampled nodes, and then leverages serial correlation to infer node representations. To ensure the quality of random sequence sampling, many efforts [27], [28] have been dedicated to incorporating graph element information such as edge weight and graph structure. For example, Line [27] utilizes first-order and second-order similarity based on edge weights, and node2vec [28] adopts Breadth-First Sampling (BFS) and Depth-First Sampling (DFS) through graph structure to guide the random wandering process. However, graph embedding methods lack generalization capabilities, which means they cannot handle dynamic graphs. In GNNs, considering the potential concerns of over-smoothing limitations [18], [19], GCN [29], a prevalent GNN model, uses first-order neighbors as receptive fields to aggregate features from neighborhood, forming an end-to-end learning paradigm. To enhance node representation power, GDC [16] introduces the network diffusion mechanism to expand the domain of the first-order receptive field in view of the sparsity of real networks. In the recommendation graph model, NGCF [12] and LightGCN [13] exploit graph structure to encode node embedding vectors to overcome the deficiencies of missing graph collaborative signals caused by direct encoding. Besides, side information such as user and service relations [30], service content [31], [32], and external knowledge graph [33], [34] has also been incorporated to improve embedding representation. However, the above GNN models rely solely on supervised signal in the model training, but fail in auto-correlation between similar node pairs or view pairs.

B. GCL-based Methods

Contrastive learning, a way of self-supervised learning, has achieved wide applications in CV, NLP and graph data mining [35]–[37]. You et al. [37] design four types of graph augmentations - node dropping, edge perturbation, attribute masking and subgraph. To our best knowledge, the existing works of recommendation are very limited based on graph contrastive learning. SGL [14] develops contrastive learning between augmented graphs by randomly sampling views, which causes uncertainty in augmented views and does not fully consider the graph structure relationship between users and services. NCL [15] performs contrastive learning between different node pairs in the graph, while ignoring high-order correlations among nodes. By analyzing the deficiencies of the above investigations, we expand the receptive field of nodes by adding a fixed global view and study the commons between multiple views through contrastive learning.

VI. Conclusion

In this paper, we propose a novel GCL-based service recommendation model, named Multi-View Graph Contrastive

Learning (MVGCL), which integrates multi-view (local and global view) with dual CL (intra-CL and inter-CL). First, we apply graph diffusion mechanism to construct a global view based on the user-service observed local view, and then combine newly generated embeddings from multiple layers and views with graph message passing. Second, we design intra-CL between each node and its first-order homogeneous neighbors, and inter-CL to attract embeddings from different views of the same node to uniformize the feature embedding space, leading to sparsity alleviation and noise denoise for improving latent feature representations of user and service nodes. Extensive experiments are conducted on three benchmark datasets and the results indicate that our proposed MVGCL model can achieve better service recommendation performance comparing with multiple state-of-the-art competing approaches.

In the future, we plan to further explore the novel strategy to generate global diffusion graph, and extend the framework to address more complex service recommendation tasks, such as sequential and POI recommendation.

Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 62272290, 62172088) and Shanghai Natural Science Foundation (No. 21ZR1400400).

References

[1] X. Xue, G. Li, D. Zhou, Y. Zhang, Z. Lu, Y. Zhao, Z. Feng, L. Cui, Z. Zhou, X. Sun, X. Lu, and S. Chen, "Research roadmap of service ecosystems: A crowd intelligence perspective," *International Journal of Crowd Science*, vol. 6, no. 4, pp. 195–222, 2022.

[2] K. Benouaret, J. Agoun, I. Benouaret, and F. Charoy, "Call limit-based composite service selection," in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 37–46.

[3] M. Li, H. Xu, Z. Tu, T. Su, X. Xu, and Z. Wang, "A deep learning based personalized QoE/QoS correlation model for composite services," in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 312–321.

[4] X. Wang, P. Zhou, Y. Wang, X. Liu, J. Liu, and H. Wu, "Servicebert: A pre-trained model for Web service tagging and recommendation," in *International Conference on Service-Oriented Computing (ICSOC)*, 2021, pp. 464–478.

[5] B. Cao, X. F. Liu, M. M. Rahman, B. Li, J. Liu, and M. Tang, "Integrated content and network-based service clustering and Web APIs recommendation for mashup development," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 99–113, 2020.

[6] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, "Billion-scale commodity embedding for e-commerce recommendation in alibaba," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018, pp. 839–848.

[7] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for Web-scale recommender systems," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018, pp. 974–983.

[8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.

[10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *International World Wide Web Conference (WWW)*, 2017, pp. 173–182.

[11] C. Chen, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Efficient neural matrix factorization without sampling for recommendation," *ACM Transactions on Information Systems*, vol. 38, no. 2, pp. 1–28, 2020.

[12] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2019, pp. 165–174.

[13] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2020, pp. 639–648.

[14] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2021, pp. 726–735.

[15] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *International World Wide Web Conference (WWW)*, 2022, pp. 2320–2329.

[16] J. Klicpera, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," *arXiv preprint arXiv:1911.05485*, 2019.

[17] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2019, pp. 235–244.

[18] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 3538–3545.

[19] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," *arXiv preprint arXiv:1905.10947*, 2019.

[20] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[21] L. Zhang, Y. Liu, X. Zhou, C. Miao, G. Wang, and H. Tang, "Diffusion-based graph contrastive learning for recommendation with implicit feedback," in *International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, 2022, pp. 232–247.

[22] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian *et al.*, "RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," in *ACM International Conference on Information and Knowledge Management (CIKM)*, 2021, pp. 4653–4664.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 249–256.

[24] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.

[25] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning (ICML)*, 2020, pp. 9929–9939.

[26] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014, pp. 701–710.

[27] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *International Conference on World Wide Web (WWW)*, 2015, pp. 1067–1077.

[28] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016, pp. 855–864.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[30] W. Li, L. Ni, J. Wang, and C. Wang, "Collaborative representation learning for nodes and relations via heterogeneous graph neural network," *Knowledge-Based Systems*, vol. 255, p. 109673, 2022.

[31] M. Liu, D. Yu, Z. Tu, and Z. Wang, "srVPA: A multi-domain conversational service recommendation approach," in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 170–175.

[32] G. Zou, Z. Qin, Q. He, P. Wang, B. Zhang, and Y. Gan, "DeepWSC: clustering Web services via integrating service composability into deep semantic features," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 1940–1953, 2022.

[33] L. Li, Y. Shi, K. Zhang, and Y. Ren, "A co-attention model with sequential behaviors and side information for session-based recommendation," in *IEEE International Conference on Web Services (ICWS)*, 2020, pp. 118–125.

[34] H. Zhang, Z. Feng, J. Xiao, Z. Ye, G. Fan, S. Chen, and X. Xue, "A knowledge graph based approach for apps permission recommendation,"

in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 176–181.

[35] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, 2020, pp. 1597–1607.

[36] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," *arXiv preprint arXiv:2104.08821*, 2021.

[37] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Advances in Neural Information Processing Systems (NIPS)*, 2020, pp. 5812–5823.