

Research Article

Node Importance Estimation with Multiview Contrastive Representation Learning

Li Liu ¹, Weixin Zeng ², Zhen Tan ¹, Weidong Xiao ¹ and Xiang Zhao ²

¹Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China

²Laboratory for Big Data and Decision, National University of Defense Technology, Changsha, China

Correspondence should be addressed to Weixin Zeng; zengweixin13@nudt.edu.cn

Received 5 April 2023; Revised 11 August 2023; Accepted 23 August 2023; Published 6 September 2023

Academic Editor: Riccardo Ortale

Copyright © 2023 Li Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Node importance estimation is a fundamental task in graph analysis, which can be applied to various downstream applications such as recommendation and resource allocation. However, existing studies merely work under a single view, which neglects the rich information hidden in other aspects of the graph. Hence, in this work, we propose a **Multiview Contrastive Representation Learning (MCRL)** model to obtain representations of nodes from multiple perspectives and then infer the node importance. Specifically, we are the first to apply the contrastive learning technique to the node importance analysis task, which enhances the expressiveness of graph representations and lays the foundation for importance estimation. Moreover, based on the improved representations, we generate the entity importance score by attentively aggregating the scores from two different views, i.e., node view and node-edge interaction view. We conduct extensive experiments on real-world datasets, and the experimental results show that MCRL outperforms existing methods on all evaluation metrics.

1. Introduction

Knowledge graphs (KGs) are graph-based data structures consisting of nodes and edges [1, 2]. Each node represents an “entity” and each edge represents a “relation” between two connected entities. In recent years, a lot of research has been devoted to solving problems related to graphs [3–6]. Estimating the importance of each node in a graph, network, and KG is a highly fundamental and crucial task, which is beneficial to many downstream applications, such as question answering, recommendation, web searching, and resource allocation [7–10].

For instance, Figure 1 shows an example of a movie knowledge graph, where “Suicide Squad” is a movie node, with the author node “David Ayer” and the actor node “Jared Leto” connecting to it via the edges “wrote” and “starred-in.” Each node is also associated with texts such as the biographies or movie plots. Before the two movies “Training Day” and “Suicide Squad” come to screen, we may use the information in the KG to estimate the potential popularity of

these two films. As can be observed from the figure, the movie node “Training Day” might become more popular, since it stars more popular actors and is directed by the director with higher recognition in terms of professional reviews.

Thus, a number of works have attempted to estimate the importance of nodes in a graph, which can be divided into two main categories. The first category includes classical methods such as PageRank [7] and Personalized PageRank [11]. PageRank was originally designed for estimating the importance of websites. It assumes that more important nodes may receive more links from other nodes. By counting the number and quality of edges linked to the nodes, the importance of the nodes can be roughly estimated, whereas this algorithm only considers the graph structure. Personalized PageRank improves PageRank by taking into account the user’s estimation of the importance of the nodes in the graph, while it neglects the type of edges. In summary, this category of approaches cannot perform well when estimating the importance of nodes in large-scale

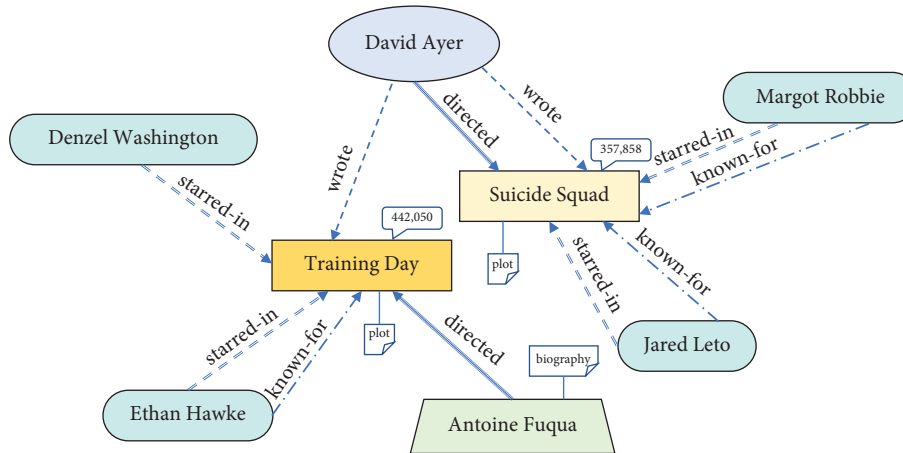


FIGURE 1: An example of movie knowledge graph. Different colored boxes represent different types of nodes (e.g., movie and director), and different edge types represent different types of relations (e.g., wrote and starred-in). The numbers connected to the movies are the known number of votes (which can be used as a source of importance scores). The “plot” and “biography” are descriptions of the nodes.

complicated graphs, since they merely take into account the graph’s topology while overlook the substantial amount of semantic and latent structural information encoded in the graph.

Another category is machine learning-based strategies, such as GENI [12] and RGTN [13]. GENI acquires the node features through node2vec and then converts them to the importance scores. The scores are flexibly aggregated via a predicate-aware attention method. Additionally, a centrality adjustment module is used at the end for fine-tuning. RGTN considers both structural and semantic information based on representation learning and then uses a common attention fusion mechanism to interact structural features with semantic features; after that, these features are projected into importance values separately and then aggregated with attention weights to produce the final importance scores of nodes. These trainable methods outperform the traditional solutions owing to the advanced supervised learning framework and the flexible graph attention mechanism.

Nevertheless, there are still notable issues with current works:

- (1) To learn the graph representations, existing efforts usually adopt a single graph learning model to capture the structural information and generate the embeddings, which can be insufficient for accurately estimating the node importance.
- (2) To aggregate the embeddings and produce the scores, existing efforts directly combine the node scores and edge embeddings, which might fail to make full use of the interaction between node and edge representations.

To tackle the aforementioned issues, we propose a Multiview Contrastive Representation Learning (MCRL) model to generate entity representations from multiple perspectives, which can help make more accurate entity importance estimation. Specifically, we adopt two graph encoders to characterize the entity representations in

different views and make cross-view contrasting to extract more useful signals using the contrastive learning strategy. Then, given the learned graph embeddings, we estimate the entity scores from two views—one based purely on the entity embeddings, and one based on the interaction between entity and relation embeddings. Finally, the multiview scores are integrated using the attention mechanism. Comprehensive experiments on real-world knowledge graph datasets validate that MCRL can outperform existing methods in terms of all metrics.

1.1. Contribution. The main contributions are summarized as follows:

- (1) We devise a multiview contrastive learning strategy to estimate entity importance, where the graph representations are first learned in the two views separately and then forwarded to the cross-view contrasting module to further enhance the expressiveness.
- (2) Based on the graph embeddings, we generate the entity importance score by attentively aggregating the scores in two views—one merely considering the entity embeddings and one modeling the interactions between entity and relation embeddings.
- (3) We conduct extensive experiments on real-world public datasets, and the results demonstrate that MCRL outperforms the baselines in all aspects.

1.2. Organization. The rest of this paper is organized as follows. Section 2 gives an overview of the literature that is relevant to this work. Section 3 provides the definitions of key concepts and the problem formulation. Section 4 elaborates the node importance estimation model. Section 5 reports and discusses the evaluation results on the mainstream node importance estimation experimental settings. Section 6 concludes the paper and provides future directions.

2. Related Work

This section provides an overview of the literature related to this work, including node importance estimation methods, graph neural network methods, contrastive learning methods, and data augmentation methods.

2.1. Node Importance Estimation. There are numerous ways to estimate node importance. PageRank (PR) [7] is a random walk model that propagates the importance of each node by traversing the graph structure or transmitting it to random nodes with a fixed probability. Personalized PageRank (PPR) [11] adjusts node weights or edge weights to bias the random walk by considering specific topics. Recently, several works have begun to explore supervised machine learning algorithms with the development of deep learning of graph data. In addition to employing the random walk model, HAR [14] additionally distinguishes between different types of predicates in KGs while being aware of importance scores to make better use of the rich information contained in KGs. GENI [12] is the earliest work to apply GNN to node importance estimation, which classifies the neighbors of each node based on the type of edges and aggregates the importance values of neighboring nodes. Additionally, it adjusts the node importance in accordance with the nodes' degree of centrality. RGTN [13] proposes a representation learning-based framework that utilizes both graph topology and node semantic information and aggregates them via an attention mechanism, which in turn infers node importance scores.

Notably, unlike previous works that use a single graph learning model to extract the structural information and generate the embeddings, which may not be adequate for estimating the node importance precisely, in this work, the graph representations are initially learned in the two views independently before being submitted to the cross-view contrasting module to further improve expressiveness. Besides, existing studies directly combine the node scores and edge embeddings, which may not fully exploit the interaction between node and edge representations; in this work, we calculate the attention scores between nodes using the representations of nodes and edges, and we then combine the scores to predict the node importance scores while fully accounting for the information contained in nodes and edges.

There are also some research works performed on heterogeneous graphs. MultiImport [15] is an end-to-end framework that integrates information from both the KG and external signals, while dealing with challenges arising from the simultaneous use of multiple input signals, such as inferring node importance from sparse signals, and potential conflicts among them. HIVEN [16] traces the local information of each node, employs the meta schema to alleviate the problem of node type dominance, and exploits the node similarity within each node type to overcome the limitation of GNN models in capturing global information. Taking the movie dataset as an example, our model requires only one type of label, the popularity of the movie, and just

ranks the importance of the movie nodes. In the work on heterogeneous graph, they also input information about other types of nodes, such as the director's box office.

2.2. Graph Neural Networks. Graph neural networks (GNNs) apply deep learning ideas to graph data, and these methods have attracted great research attention in recent years [17–19]. The pioneering work of GNN is the graph convolution model GCN [20], which performs convolution in the Fourier domain by aggregating neighbor node features and has performed well in many applications. However, GCN training needs to use the neighbor matrix of the whole graph, which depends on the specific graph structure, so GraphSAGE [21] is proposed to solve this problem. GraphSAGE uses a multilayer aggregation function, and each layer of aggregation function will aggregate the information of nodes and their neighbors to get the feature vector of the next layer, which uses the neighborhood information of nodes and does not depend on the global graph structure. In addition, GCN treats all neighboring nodes equally in convolution and cannot assign different weights to nodes according to their importance; graph attention networks (GATs) [22] are proposed to solve this problem. GATs adaptively aggregate neighboring information based on the attention mechanism and can assign different weights to different nodes, and they provide an efficient framework for integrating deep learning into graph mining. These GNN works have been widely used in recommender systems [23], knowledge graph inference [24], and graph classification [25].

2.3. Contrastive Learning on Graphs. Contrastive learning (CL) has recently become recognized as an effective method for learning self-supervised graph representations [26–29]. CL can produce data representations by learning to encode similarities or dissimilarities between a set of unlabeled samples. Rich unlabeled data are used as a supervised signal for model training. Since there are typically few labeled entities in knowledge graphs, in this work, we employ contrastive learning and make use of a large number of unlabeled entities to better obtain feature representations of nodes and get more precise node importance scores.

3. Problem Formulation

In this section, we provide the definitions of key concepts and introduce the formalization of the problem studied in our work.

3.1. Knowledge Graph. A knowledge graph is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$ that represents a network of real-world entities and illustrates the relationship between them, where \mathcal{V} , \mathcal{E} , and \mathcal{P} represent the entities, relationships, and predicates, respectively. In the knowledge graph, it is plausible that there might be several different types of predicates between two entities, and hence each edge is linked to a particular predicate through a mapping function: $\mathcal{E} \rightarrow \mathcal{P}$.

3.2. Node Importance. An entity's importance or popularity in a knowledge graph is indicated by node importance $s \in R$, which is a nonnegative real value.

3.3. Semantic Information of Nodes. The semantic information of the node is the natural language text that provides comprehensive descriptions of the semantic information of the entity or the concept represented by the node.

3.4. Problem Definition. Given a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$, a set of semantic information of nodes \mathcal{T} , and importance scores \mathcal{S} for a subset of nodes $\mathcal{V}_s \subset \mathcal{V}$, entity importance estimation aims to learn a function $f: \mathcal{V} \rightarrow [0, \infty)$ that generates the importance score for each node in the knowledge graph.

4. Approach

In this section, we first describe the outline of the proposed model. Then, we introduce the details of the two components and training. Table 1 provides the definition of symbols used in this paper.

4.1. Outline. As shown in Figure 2, the features of entities are first forwarded to the self-supervised contrastive learning module to generate node embeddings. By adjusting the encoders' hidden-size and out-size parameters, we produce a high-dimensional embedding and a low-dimensional embedding for each node, respectively. Then, the high-dimensional embeddings are mapped directly to the importance score (i.e., score 1), and the low-dimensional embeddings are concatenated with the edge features, which are mapped to another importance score (i.e., score 2) using the attention mechanism. The reason for this is that node embedding with high dimensionality can save more information and is better suited for direct mapping to predict the importance score; node embedding with low dimensionality can be better combined with edge embedding to get the attention weight between two nodes, and then the scores are aggregated to make the prediction. The scores from the two perspectives are combined to obtain the final predicted node importance score. Finally, we train the entire model by aggregating the self-supervised contrastive loss, the supervised root mean square error (RMSE) loss, and the learning-to-rank (LTR) loss.

4.2. Multiview Contrastive Learning. In this work, we choose two popular GNN models as the encoders to produce the graph representations in different views.

4.2.1. GCN. The first one is the graph convolutional network (GCN) [20]. The GCN model utilizes multiple convolutional layers to conduct the information passing by aggregating the features of nodes and their first-order neighborhoods. Given two message passing layers, the equation of GCN can be expressed as follows:

$$\mathbf{E} = \sigma(\tilde{\mathbf{A}}\sigma(\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}_0)\mathbf{W}_1), \quad (1)$$

where \mathbf{X} is the input feature vector matrix of nodes, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I} \in \mathbf{R}^{N \times N}$ is the scaled adjacency matrix \mathbf{A} of the graph \mathcal{G} with added self-loops, \mathbf{W}_0 and \mathbf{W}_1 are trainable weight matrixes, σ is the activation function, and \mathbf{E} is the output node embedding matrix.

4.2.2. GAT. The other model is graph attention network (GAT) [22], which assigns attention weight coefficients to the neighboring nodes of the target node and uses a local aggregation function to generate node embeddings. Given the features of nodes, the influence of node j on node i can be calculated by the following equation:

$$\alpha_{ij} = \frac{\exp(\sigma(\mathbf{v}[\mathbf{W}\mathbf{e}(i) \parallel \mathbf{W}\mathbf{e}(j)]))}{\sum_{k \in \mathcal{N}(i)} \exp(\sigma(\mathbf{v}[\mathbf{W}\mathbf{e}(i) \parallel \mathbf{W}\mathbf{e}(k)]))}, \quad (2)$$

where $\mathbf{e}(\cdot)$ represents the feature of the node, \parallel is a concatenation operator, σ is the activation function, $\mathcal{N}(i)$ represents the neighboring nodes of node i , \mathbf{v} is a learnable weight vector, and \mathbf{W} is a trainable weight matrix.

Following the acquisition of the attention weights, GAT aggregates the feature representations of the nodes and its neighbors. The L -layer GAT aggregation formulation with multihead attention can be expressed as follows:

$$\mathbf{e}_i^{(k+1)} = \parallel_{h=1}^H \sigma \left(\sum_{j \in \mathcal{N}(i)} (\alpha_{ij}^{(k)} \mathbf{W}^{(k)})^h \mathbf{e}_j^{(k)} \right), k = 1, 2, \dots, L, \quad (3)$$

where \mathbf{e}_i^0 is the input feature vector matrix of nodes in the graph.

4.2.3. Cross-View Contrastive Learning. After obtaining the representations in the two views, we use the cross-view contrastive learning strategy to help learn more expressive graph representations.

Given a node, we denote its embedding generated by the first view as $\mathbf{e}_\varphi(i)$ and the embedding generated by the second view as $\mathbf{e}_\psi(i)$. These two embeddings form a positive sample. The pairs of embeddings including $\mathbf{e}_\varphi(i)$ (or $\mathbf{e}_\psi(i)$) and another node's embedding are the negative samples. The following is the definition of node i 's contrastive object:

$$\begin{aligned} \mathcal{L}^\varphi(i) &= -\log \frac{\exp(f(\mathbf{e}_\varphi(i), \mathbf{e}_\psi(i)))}{\exp(f(\mathbf{e}_\varphi(i), \mathbf{e}_\psi(i))) + \mathcal{N}_{\text{cross}} + \mathcal{N}_{\text{intra}}}, \\ \mathcal{N}_{\text{cross}} &= \sum_{k=1}^n 1_{[k \neq i]} \exp(f(\mathbf{e}_\varphi(i), \mathbf{e}_\psi(k))), \\ \mathcal{N}_{\text{intra}} &= \sum_{k=1}^n 1_{[k \neq i]} \exp(f(\mathbf{e}_\varphi(i), \mathbf{e}_\varphi(k))), \end{aligned} \quad (4)$$

where f is a score function that measures the similarity between two embeddings. Specifically, two embeddings

TABLE 1: Table of symbols.

Symbol	Definition
\mathbf{X}	The input feature vector matrix of nodes
\mathbf{A}	The scaled adjacency matrix of the graph \mathcal{G}
$\mathbf{W}, \mathbf{W}_0, \mathbf{W}_1$	Trainable weight matrixes
σ	Activation function
$\mathbf{e}(\cdot)$	Feature of the node or edge
$\mathcal{N}(i)$	Neighboring nodes of node i
f	Score function that measures the similarity between two embeddings
$\mathbf{1}_{[\cdot]}$	Indicator function
$\mathcal{N}_{\text{cross}}$	Cross-view negative samples
$\mathbf{e}_\varphi(i), \mathbf{e}_\psi(i)$	Embeddings of node i generated by first view and the second view
α_{ij}	Attention weight of node i to node j
$s_1(i), s_2(i)$	Predicted scores of node i of node view and node-edge interaction view
\mathcal{V}_s	Set of nodes with known importance scores
\mathbf{E}	Output node embedding matrix of GCN
$\tilde{\mathbf{A}}$	\mathbf{A} with added self-loops
\mathbf{v}	Learnable weight vector
\parallel	Concatenation operator
$\mathbf{e}_i(k)$	Embedding of node i in the layer k of GAT
n	Number of nodes in the graph
F	Fully connected neural network
a	Hyperparameter
$\mathcal{N}_{\text{intra}}$	Intraview negative samples
P_{ij}	Predicate between nodes i and j
$s(\cdot)$	Valid ground truth importance value of node
$s^*(i)$	Predicted score of node i

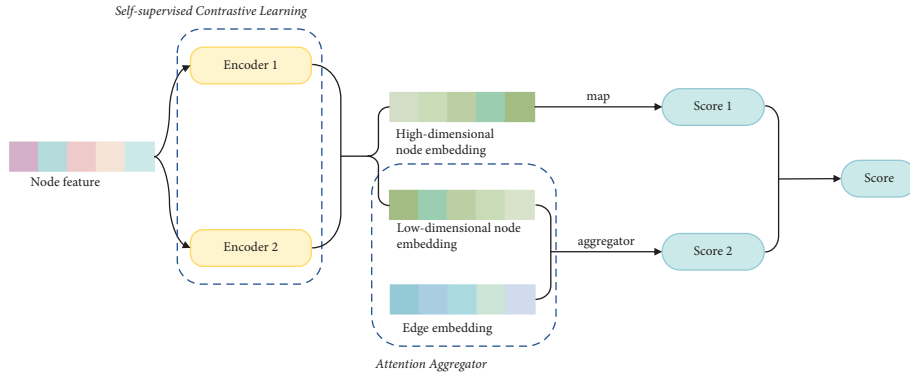


FIGURE 2: The framework of our proposed MCRL.

are first transformed using a multilayer perceptron (MLP) with nonlinear activation functions, and then the similarity between the two embeddings is evaluated by using the similarity metric. n is the number of nodes in the graph, and $\mathbf{1}_{[\cdot]}$ is an indicator function that returns 1 if the argument included in the bracket holds true and 0 otherwise. The first term in the denominator is the positive sample. The term $\mathcal{N}_{\text{cross}}$ refers to the cross-view negative samples and $\mathcal{N}_{\text{intra}}$ represents the intraview negative samples.

Finally, the overall self-supervised loss is defined as

$$\mathcal{L}_c = \frac{1}{2n} \sum_{i=1}^n [\mathcal{L}^\varphi(i) + \mathcal{L}^\psi(i)]. \quad (5)$$

4.3. Multiview Score Aggregation. We devise a multiview strategy to produce and aggregate the entity importance scores.

4.3.1. Node View. As the two encoders obtain the high-dimensional embeddings of each node i in the graph, respectively, i.e., $\mathbf{e}_\varphi(i)$ and $\mathbf{e}_\psi(i)$, we add the two embeddings and generate the node importance scores of the node view:

$$s_1(i) = F(\mathbf{e}_\varphi(i) + \mathbf{e}_\psi(i)), \quad (6)$$

where F represents a fully connected neural network in our experiments.

4.3.2. Node-Edge Interaction View. The node view merely focuses on the features of nodes. However, edge features also contain a wealth of information and play an essential role in terms of estimating node importance. Thus, we concatenate the node and edge vectors to better model their interactions. Specifically, we use the attention mechanism, and the attention weight of node i to node j can be calculated by the following equation:

$$\alpha_{ij} = \frac{\exp\left(\sigma\left(\mathbf{v}\left[\mathbf{e}(i)\parallel\mathbf{e}(p_{ij})\parallel\mathbf{e}(j)\right]\right)\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(\sigma\left(\mathbf{v}\left[\mathbf{e}(i)\parallel\mathbf{e}(p_{ij})\parallel\mathbf{e}(j)\right]\right)\right)}, \quad (7)$$

where $\mathbf{e}(\cdot)$ represents the feature of the node (or edge), p_{ij} denote the predicate between nodes i and j , \parallel is a concatenation operator, σ is the activation function, $\mathcal{N}(i)$ represents the neighboring nodes of node i , and \mathbf{v} is a learnable weight vector.

We first convert the low-dimensional embeddings to scores, and the scores can then be aggregated using the attention weights obtained with the following formulation:

$$s_2(i) = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij} F(\mathbf{e}_\varphi(i) + \mathbf{e}_\psi(i)). \quad (8)$$

4.3.3. Score Aggregation. The final predicted scores are formed by integrating the scores from the node view and the node-edge interaction view:

$$s^*(i) = as_1(i) + (1-a)s_2(i), \quad (9)$$

where a is the hyperparameter.

4.4. Training. We select mean square error and learning-to-rank loss as supervised loss functions. First, we establish the node set \mathcal{V}_s using nodes with known importance ratings. The following equation illustrates how to utilize RMSE to calculate the error between the predicted and labeled nodes' important scores:

$$\mathcal{L}_e = \frac{1}{|\mathcal{V}_s|} \sum_{i \in \mathcal{V}_s} (s^*(i) - s(i))^2, \quad (10)$$

where $s(i)$ is the valid ground truth importance value of node i and $s^*(i)$ represents the predicted score.

In order to take the entire graph into account while rating the nodes' importance, we use the learning-to-rank (LTR) loss in the training process. We sample n nodes for node i to form a node set \mathcal{N}_i^n . The calculation method is shown below:

$$\begin{aligned} \hat{s}(i) &= \frac{\exp(s(i))}{\sum_{j \in \mathcal{N}_i^n} \exp(s(j))}, \\ \hat{s}^*(i) &= \frac{\exp(s^*(i))}{\sum_{j \in \mathcal{N}_i^n} \exp(s^*(j))}, \\ \mathcal{L}_r(i) &= - \sum_{i \in \mathcal{N}_i^n} \hat{s}(i) \log(\hat{s}^*(i)). \end{aligned} \quad (11)$$

By combining the supervised loss function with the self-supervised comparison loss function, we can obtain the total loss function for model training:

$$\mathcal{L} = \mathcal{L}_e + \frac{1}{|\mathcal{V}_s|} \sum_{j \in \mathcal{V}_s} \mathcal{L}_r(j) + \mathcal{L}_c. \quad (12)$$

5. Experiments

In this section, we conduct extensive experiments on real-world datasets to answer the following questions:

- (1) Does MCRL work better than the existing baseline and previous models? Are the contrastive learning and score aggregation modules useful?
- (2) Is MCRL generally valid for different encoders? Is it sensitive to hyperparameters?

We describe detailed information about the dataset and baseline in Section 5.1, answer the above questions in Sections 5.2 to 5.4, and perform a case study in Section 5.5.

5.1. Experimental Setting

5.1.1. Datasets. Following previous works, we conduct comprehensive experiments on three public knowledge graphs with different features. More details can be found in Table 2.

FB15K [30] is a subset of the Freebase [31] database which contains knowledge base relation triples and textual mentions of entity pairs. The 30-day view count of the corresponding Wikipedia page is utilized as the node importance score for each entity in the graph, and the description of the entity in the Wikidata is used as the node semantic information. Compared to the rest of the datasets, FB15K has more predicates and also a higher density.

TMDB5K is a movie knowledge graph generated from TMDB (<https://www.kaggle.com/tmdb/tmdb-movie-metadata>), and it contains information about movies as well as other closely related entities including actors, casts, crews, and countries. The popularity of the movie is conducted to identify the entity's importance scores, while the movie summaries provide the nodes' semantic information.

IMDB is a movie knowledge graph created from the IMDB dataset (<https://www.imdb.com/interfaces/>), which includes entities for movies, casts, crews, genres, publishing companies, and countries. The importance scores are determined by the number of votes for each movie. As the semantic information for the nodes, the movie plot summaries and personal biographies are used.

5.1.2. Competing Methods. We compare MCRL with two primary kinds of methods that are readily available for ranking the importance of nodes in the graph. The first refers to the unsupervised approaches, including

- (1) PR [7]: a random walk-based algorithm for measuring the importance of web pages can also be used to rank the importance of nodes in a graph.

TABLE 2: Summary of the datasets used for evaluation.

Dataset	Nodes	Edges	Edge types	Nodes with values	Importance value type
FB15K	14,951	592,213	1,345	14,105 (94%)	Pageviews
TMDB5K	114,805	761,648	34	4,803 (4%)	Movie popularity
IMDB	1,124,995	9,729,868	30	202,538 (18%)	Vote number for movie

- (2) PPR [11]: a variant of PageRank that considers the node’s own feature information.

The second includes the supervised methods:

- (1) LR: a simple machine learning technique that uses the least squares algorithm based on the reduction of mean square errors.
- (2) RF: another basic machine learning algorithm that uses the ensemble learning method based on decision trees.
- (3) GCN [20]: a GNN model that aggregates the neighbor node embeddings to conduct graph convolutions in the Fourier domain.
- (4) GAT [22]: a GNN model that uses the multihead attention mechanism to aggregate the features of the neighbor nodes.
- (5) GENI [12]: the model aggregates scores using a predicate-aware attention mechanism and flexible centrality adjustment to perform node importance estimation.
- (6) RGTN [13]: the model provides a representation learning-based framework for node importance estimation, which propagates the embedding of nodes in a relational graph transformer.

5.1.3. Detailed Settings. For fair comparison, we maintain consistency with previous works [12, 13] by concatenating semantic and structural features as node input features, except for GENI* using the structure features only by following the setting in the original paper [12]. The structural features of the nodes are obtained by node2vec [32], and the semantic features are obtained from Transformer-XL [33]. To enhance the GNN model in acquiring the node representation more accurately, we employ two widely used graph data augmentation techniques during training. Given a graph, edge dropout [34] refers to randomly dropping some edges with the probability p , while node dropout [35] is the process of randomly discarding some nodes and their connected edges with the ratio q . The nodes with the important scores in datasets are divided into training, validation, and testing parts with a ratio of 7 : 1 : 2. To obtain reliable and stable experimental results, we conducted five-fold cross-validation on each dataset to evaluate all the models. In order to avoid the overfitting problem, we apply early stopping if the performance on the validation set is not improved for 1000 consecutive epochs. For testing, the parameters that perform the best during validation are used. The experimental setup includes a Linux operating system, an NVIDIA GeForce RTX 3090 graphics card with 24 GB of memory,

CUDA version 11.3, and Python 3.8 programming language, and the model is built using PyTorch framework version 1.11.0.

5.1.4. Evaluation Metrics. Following previous works [12, 13], we use three evaluation metrics to give a thorough evaluation of the rank quality and importance relevance: normalized discounted cumulative gain (NDCG) [36], Spearman’s rank correlation coefficient (SPEARMAN) [37], and Top-K Hit Ratio (HR). For all metrics, higher values are preferable. The definitions of metrics in formal terms are provided below.

- (1) NDCG is a popular metric for evaluating ranking quality for the top k nodes. Given a list of k nodes ranked by predicted important scores and their ground truth important scores $s(i)$, the discounted cumulative gain at position k (DCG@ k) can be defined. The Ideal DCG at rank position k (IDCG@ k) is obtained by an ideal ordering of nodes based on their ground truth scores. Then, we can get the normalized DCG at position k (NDCG@ k).
- (2) SPEARMAN measures the strength and direction of the correlation between two node rankings that are rated in accordance with the predicted scores $s^*(i)$ and the ground truth scores $s(i)$.
- (3) HR measures the ratio of the predicted nodes that have been contained by the real important nodes. HR@ k is achieved through $HR@k = \text{NumberofHits}@k/k$.

5.2. Analysis of Experimental Results. The performance results are shown in Table 3. Numbers after \pm symbol refer to standard deviation from the cross-validation. The approach denoted by an asterisk (*) only employs structure features.

It can be observed from the table that our proposal MCRL outperforms all the compared models in all metrics. Besides, the results also reveal the following:

- (1) Supervised methods typically perform better than unsupervised approaches and are more accurate in predicting node importance scores.
- (2) GENI prematurely maps node features into scores and calculates attention weights by simply splicing node scores with edge embeddings, which cannot fully utilize the interaction between nodes and edges.
- (3) RGTN learns graph representations from a single perspective, which is not flexible and accurate enough for node features.
- (4) The model proposed in this paper also has some shortcomings because it uses data augmentation methods that randomly discard some nodes or edges

TABLE 3: Experimental results on real-world datasets.

Method	FB15K			TMDB5K			IMDB		
	NDCG @100	SPEARMAN	HR @100	NDCG @100	SPEARMAN	HR @100	NDCG @100	SPEARMAN	HR @100
PR	0.8395 ± 0.009	0.3505 ± 0.018	0.1340 ± 0.015	0.8378 ± 0.016	0.6092 ± 0.012	0.4140 ± 0.064	0.8485 ± 0.032	0.1829 ± 0.003	0.3720 ± 0.017
PPR	0.8407 ± 0.009	0.3656 ± 0.022	0.1360 ± 0.019	0.8594 ± 0.007	0.7248 ± 0.013	0.4220 ± 0.060	0.8638 ± 0.029	0.3979 ± 0.004	0.4060 ± 0.019
LR	0.8921 ± 0.011	0.6097 ± 0.017	0.2160 ± 0.042	0.8443 ± 0.015	0.6871 ± 0.011	0.4300 ± 0.034	0.8972 ± 0.003	0.5838 ± 0.003	0.4540 ± 0.024
RF	0.9136 ± 0.010	0.6441 ± 0.009	0.2220 ± 0.048	0.8617 ± 0.014	0.6993 ± 0.016	0.4680 ± 0.028	0.9068 ± 0.006	0.6112 ± 0.004	0.4380 ± 0.018
GCN	0.9408 ± 0.007	0.7179 ± 0.013	0.4240 ± 0.051	0.8999 ± 0.008	0.7736 ± 0.011	0.5240 ± 0.029	0.9031 ± 0.009	0.7135 ± 0.006	0.3240 ± 0.035
GAT	0.9357 ± 0.014	0.6898 ± 0.047	0.4060 ± 0.055	0.9017 ± 0.010	0.7728 ± 0.008	0.5360 ± 0.031	0.9186 ± 0.008	0.6942 ± 0.007	0.4780 ± 0.028
GENI*	0.9301 ± 0.005	0.7385 ± 0.012	0.4260 ± 0.092	0.9018 ± 0.006	0.7898 ± 0.009	0.5460 ± 0.024	0.9302 ± 0.005	0.7312 ± 0.005	0.5080 ± 0.034
GENI	0.9415 ± 0.006	0.7797 ± 0.018	0.4220 ± 0.058	0.8979 ± 0.008	0.7721 ± 0.007	0.5340 ± 0.021	0.9394 ± 0.005	0.7351 ± 0.005	0.5420 ± 0.026
RGTN	<i>0.9501 ± 0.007</i>	<i>0.8156 ± 0.010</i>	<i>0.4880 ± 0.064</i>	<i>0.9114 ± 0.009</i>	<i>0.7946 ± 0.010</i>	<i>0.5580 ± 0.036</i>	<i>0.9585 ± 0.004</i>	<i>0.7643 ± 0.003</i>	<i>0.5660 ± 0.038</i>
HIVEN	0.9436 ± 0.006	0.7673 ± 0.015	0.4280 ± 0.052	0.9045 ± 0.008	0.7852 ± 0.008	0.5480 ± 0.026	0.9388 ± 0.006	0.7475 ± 0.006	0.5340 ± 0.036
MCRL	0.9626 ± 0.013	0.8229 ± 0.014	0.5100 ± 0.069	0.9195 ± 0.020	0.7961 ± 0.013	0.5680 ± 0.041	0.9624 ± 0.008	0.7713 ± 0.007	0.5720 ± 0.044

The bolded results are the best, and the italic results are the second best.

in the graph, thus increasing the uncertainty and instability, and the standard deviations from the cross-validation are a little bit higher.

5.3. Ablation Study. In this section, we perform ablation studies to prove the validity of each module in our proposed framework.

5.3.1. On Multiview Contrastive Learning. To verify the effectiveness of contrastive learning, we conducted ablation study on two datasets. Specifically, one variant merely uses GCN as the encoder and the other merely uses GAT as the encoder, both removing the contrastive loss component in the training. It can be observed from Figure 3 that using contrastive learning effectively increases the performance by improving the graph representations.

5.3.2. On Multiview Score Aggregation. To validate the performance of the score aggregation module and also to demonstrate that splicing the features of nodes with those of edges to calculate attention weights is more effective than simply splicing the scores of nodes with the features of edges, we conduct the experiment on multiview score aggregation. Table 4 shows that employing score aggregation enhances the effectiveness and stability of model prediction. Besides, concatenating the node and edge representations can better capture their interactions and generate superior performance.

5.4. Further Experiments

5.4.1. Choices of Encoders. In this study, we select the GCN and GAT models as the encoders and employ contrastive learning approaches to produce the node representations. In fact, the encoders in the model can be replaced with any graph representation learning model, and we choose GraphSAGE for experiments for verification.

From Figure 4, we can see that the outcomes of the experiments on the two datasets are not sensitive to the choices of encoders, and employing contrastive learning can consistently enhance the performance. Therefore, MCRL can be applied to a variety of encoders, and in this paper, we choose two popular encoders.

5.4.2. Parameter Sensitivity. As mentioned in Section 4.3, we obtain the final prediction scores by assigning hyperparameter a as the weight to the scores from two perspectives. To show that the model is stable under hyperparameter perturbations, we conduct sensitivity analyses on this important hyperparameter on FB15K. Figure 5 demonstrates that changing the weights of the two scores has no appreciable impact on the experimental outcomes. Therefore, MCRL is robust to the perturbations of a .

5.4.3. Training Time. To compare the efficiency of different methods, we report the overall training time on the FB15K

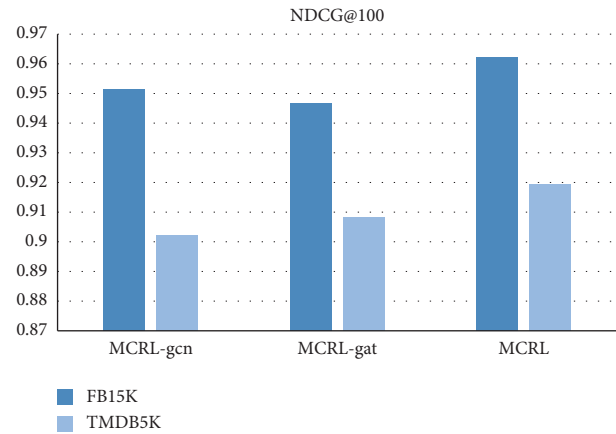


FIGURE 3: The results of employing two encoders for contrastive learning versus using just one encoder. MCRL-gcn: the model only uses GCN as the encoder. MCRL-gat: the model only uses GAT as the encoder. MCRL: our proposed model, which uses two encoders and employs contrastive learning.

dataset in Table 5. For each model, we ran the experiment five times and took the average of the run times to obtain the time cost. As shown in Table 5, it is evident that contrastive learning and dropout method used in our proposed model increase the model training time, while the slight increase in training time is acceptable considering the improvement in prediction results, which is shown in Table 3.

5.4.4. Comparison with Methods Proposed for Heterogeneous Graphs. Recent years have also witnessed the emergence of importance estimation methods on heterogeneous graph. Thus, for the comprehensiveness of experiment, we compare our proposal with state-of-the-art importance estimation method on heterogeneous graph, i.e., HIVEN [16], and report the performance in Table 3. In order to compare the outcomes of the two models without taking into account the different node types, we apply HIVEN on homogeneous graph with the same input data as our proposed model. Taking the movie dataset as an example, we only provide the models with labels of some of the movie nodes to just estimate the importance of the movie nodes. The experimental results demonstrate that our work outperforms the method proposed for heterogeneous graphs when evaluating on the homogeneous graph. This indicates that the methods proposed for heterogeneous graphs cannot work well on homogeneous graphs where the nodes are of the same type.

5.5. Case Study Analysis. To demonstrate the effectiveness of MCRL on the prediction task, we conduct a case study using the movie dataset IMDB as an example. Table 6 shows the top-10 movies with the highest importance scores predicted by MCRL, GENI, and RGTN along with the difference between their ground truth ranks and estimated ranks. The ground truth rank is calculated from known importance scores of movies. From the table, we can see that the top-10

TABLE 4: The results of MCRL and its variants.

Data	Metric	MCRL-none	MCRL-score	MCRL
FB15K	NDCG@100	0.9485 ± 0.034	0.9502 ± 0.015	0.9626 ± 0.013
	SPEARMAN	0.7901 ± 0.026	0.8173 ± 0.012	0.8229 ± 0.014
	HR@100	0.4829 ± 0.094	0.4902 ± 0.075	0.5100 ± 0.069
TMDB5K	NDCG@100	0.9054 ± 0.048	0.9107 ± 0.021	0.9195 ± 0.020
	SPEARMAN	0.7768 ± 0.022	0.7826 ± 0.016	0.7961 ± 0.013
	HR@100	0.5427 ± 0.069	0.5579 ± 0.038	0.5680 ± 0.041
IMDB	NDCG@100	0.9422 ± 0.016	0.9535 ± 0.007	0.9624 ± 0.008
	SPEARMAN	0.7593 ± 0.014	0.7664 ± 0.009	0.7713 ± 0.007
	HR@100	0.5545 ± 0.067	0.5636 ± 0.038	0.5720 ± 0.044

MCRL-none: the model without score aggregation module. MCRL-score: the model that simply splices the scores of nodes with the features of edges to calculate attention weights. MCRL: our proposed model, which splices the features of nodes with those of edges to calculate attention weights.

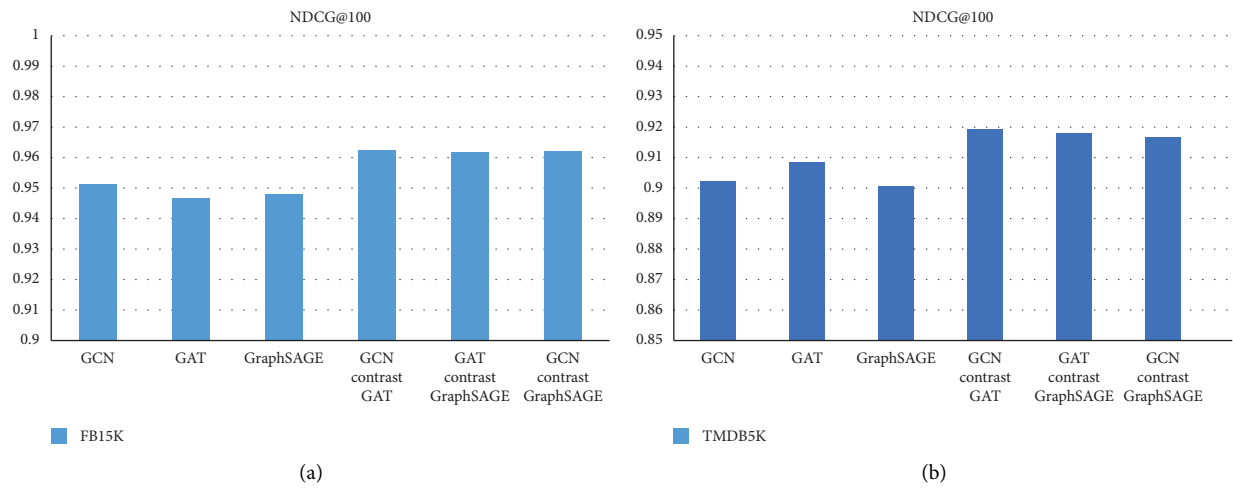


FIGURE 4: Results of experiments with different encoders on (a) FB15K and (b) TMDB5K.

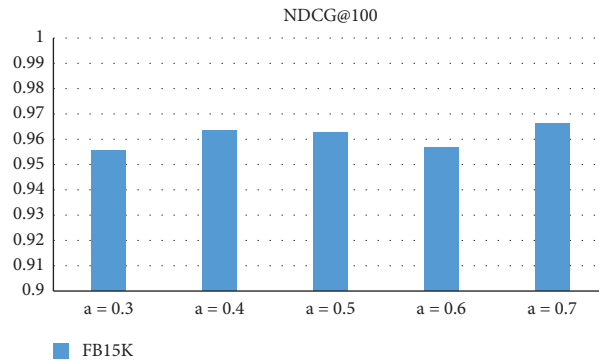


FIGURE 5: Results of experiments with different hyperparameters on FB15K.

TABLE 5: Training time comparison of existing methods.

Method	Training time (min)
GCN	1
GAT	3
GENI*	4
GENI	5
RGTN	1
HIVEN	3
MCRL	8

TABLE 6: Top-10 movies with highest predicted importance scores by MCRL, GENI, RGTN, and HIVEN on IMDB.

Ranking	MCRL	G-E	GENI	G-E	RGTN	G-E	HIVEN	G-E
1	Batman Begins	16	Cattle King	49	Shrek	145	Alice in Wonderland	351
2	Avengers: Endgame	75	Hercules	880	I Am Number Four	760	Magnolia	544
3	Finding Nemo	45	Aladdin	368	I Am Legend	111	A History of Violence	776
4	The Intouchables	83	Titanic	32	The Avengers	14	The Princess Bride	334
5	Transformers	139	Tarzan	892	Star Trek Beyond	767	Oldeuboi	211
6	Star Wars: Episode VII-The Force Awakens	56	Spider-Man	107	Godzilla	360	Iron Man Three	79
7	Guardians of the Galaxy	26	The Day After Tomorrow	311	Sherlock Holmes	158	The Lincoln Lawyer	836
8	Inception	-5	Jerry Maguire	685	Deja Vu	525	The Usual Suspects	37
9	The Avengers	9	Spider-Man 2	184	The Matrix	-2	Mr. Brooks	1312
10	The Hunger Games: Catching Fire	134	Minority Report	210	Baby Driver	277	Prisoners	148

The value "G-E" of "ground truth rank" minus "estimated rank" is shown for each prediction.

movies predicted by MCRL are qualitatively better than the two others, demonstrating our model's effectiveness in terms of evaluation.

6. Conclusion

Estimating the importance of nodes in KGs is a highly fundamental and crucial task in graph analysis, which is beneficial to many downstream applications. In this paper, we propose a multiview contrastive learning strategy to obtain representations of nodes from multiple perspectives and use cross-view contrasting module to enhance the expressiveness. Additionally, we generate the entity importance score by attentively aggregating the scores in two views—one merely considering the entity embeddings and one modeling the interactions between entity and relation embeddings. Comprehensive experiments on real-world knowledge graphs show that our model outperforms existing methods in measures. There are also some works on node importance estimation for heterogeneous graphs [15, 16], so for future work, we intend to apply cutting-edge representation learning techniques to estimate node importance on heterogeneous knowledge graphs.

Data Availability

This study used the movie datasets from TMDb and IMDb, the TMDb dataset can be downloaded from website <https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata>, and the IMDb dataset can be downloaded from the official IMDb website <https://www.imdb.com/interfaces/or> <https://datasets.imdbws.com/>. The datasets are available for access to customers for personal and noncommercial use.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Key R&D Program of China No. 2022YFB3102600, NSFC under grant nos. 62302513 and 62272469.

References

- [1] A. Hogan, E. Blomqvist, M. Cochez et al., "Knowledge graphs," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–37, 2021.
- [2] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2022.
- [3] W. Zeng, X. Zhao, Z. Tan, J. Tang, and X. Cheng, "Matching knowledge graphs in entity embedding spaces: an experimental study," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023, <https://ieeexplore.ieee.org/document/10114983>.
- [4] W. Zeng, X. Zhao, X. Li, J. Tang, and W. Wang, "On entity alignment at scale," *The VLDB Journal*, vol. 31, no. 5, pp. 1009–1033, 2022.
- [5] Y. Fang, X. Zhao, P. Huang, W. Xiao, and M. de Rijke, "Scalable representation learning for dynamic heterogeneous information networks via metagraphs," *ACM Transactions on Information Systems*, vol. 40, no. 4, pp. 1–27, 2022.
- [6] Y. Fang, X. Zhao, Y. Chen, W. Xiao, and M. de Rijke, "PF-HIN: Pre-Training for Heterogeneous Information Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8372–8385, 2023.
- [7] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical report, Stanford University, Stanford, CA, USA, 1998.
- [8] S. P. Borgatti, "Centrality and network flow," *Social Networks*, vol. 27, no. 1, pp. 55–71, 2005.
- [9] T. Crnovrsanin, I. Liao, Y. Wu, and K. L. Ma, "Visual recommendations for network navigation," *Computer Graphics Forum*, vol. 30, no. 3, pp. 1081–1090, 2011.
- [10] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, "Pagerank with priors: an influence propagation perspective," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence Citeseer*, Beijing China, August 2013.
- [11] T. H. Haveliwala, "Topic-sensitive pagerank," in *Proceedings of the 11th International Conference on World Wide Web*, pp. 517–526, Honolulu HI USA, May 2002.
- [12] N. Park, A. Kan, X. L. Dong, T. Zhao, and C. Faloutsos, "Estimating node importance in knowledge graphs using graph neural networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 596–606, Anchorage AK USA, August 2019.
- [13] H. Huang, L. Sun, B. Du, C. Liu, W. Lv, and H. Xiong, "Representation learning on knowledge graphs for node importance estimation," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 646–655, Singapore, August 2021.
- [14] X. Li, M. K. Ng, and Y. Ye, "HAR: hub, authority and relevance scores in multi-relational data for query search," in *Proceedings of the 2012 SIAM International Conference on Data Mining SIAM*, pp. 141–152, Anaheim, CA, USA, April 2012.
- [15] N. Park, A. Kan, X. L. Dong, T. Zhao, and C. Faloutsos, "MultiImport: inferring node importance in a knowledge graph from multiple input signals," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 503–512, New York, NY, USA, July 2020.
- [16] C. Huang, Y. Fang, X. Lin, X. Cao, W. Zhang, and M. Orłowska, "Estimating node importance values in heterogeneous information networks," in *Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 846–858, IEEE, Kuala Lumpur, Malaysia, May 2022.
- [17] S. Lu, Z. Zhu, J. M. Gorriç, S. H. Wang, and Y. D. Zhang, "NAGNN: classification of COVID-19 based on neighboring aware representation from deep graph neural network," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1572–1598, 2022.
- [18] Y. Pang, T. Huang, Z. Wang et al., "Graph Decipher: a transparent dual-attention graph neural network to understand the message-passing mechanism for the node classification," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8747–8769, 2022.
- [19] J. Bai, Y. Ren, and J. Zhang, "Measuring and sampling: a metric-guided subgraph learning framework for graph

- neural network,” *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 7502–7525, 2022.
- [20] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [21] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” 2017, <https://arxiv.org/abs/1710.10903>.
- [23] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, “Am-gcn: adaptive multi-channel graph convolutional networks,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1243–1253, California, CA, USA, July 2020.
- [24] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting,” 2017, <https://arxiv.org/abs/1709.04875>.
- [25] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, <https://arxiv.org/abs/1609.02907>.
- [26] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, “Graph contrastive learning with adaptive augmentation,” in *Proceedings of the Web Conference 2021*, pp. 2069–2080, Ljubljana Slovenia, April 2021.
- [27] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep graph infomax,” 2018, <https://arxiv.org/abs/1809.10341>.
- [28] J. Chen, Y. Wang, and G. Zhang, “Contrastive learning for graphs with locality-sensitive hashing,” *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 4, pp. 1–25, 2020.
- [29] D. Cai, S. Qian, Q. Fang, J. Hu, W. Ding, and C. Xu, “Heterogeneous graph contrastive learning network for personalized micro-video recommendation,” *IEEE Transactions on Multimedia*, vol. 25, pp. 2761–2773, 2023.
- [30] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” *Advances in Neural Information Processing Systems*, vol. 26, 2013.
- [31] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for structuring human knowledge,” in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1247–1250, Vancouver Canada, June 2008.
- [32] A. Grover and J. Leskovec, “node2vec: scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, San Francisco, CA, USA, August 2016.
- [33] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-xl: attentive language models beyond a fixed-length context,” 2019, <https://arxiv.org/abs/1901.02860>.
- [34] Y. Rong, W. Huang, T. Xu, and J. Huang, “Dropegde: towards deep graph convolutional networks on node classification,” 2019, <https://arxiv.org/abs/1907.10903>.
- [35] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, “Graph contrastive learning with augmentations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 5812–5823, 2020.
- [36] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.
- [37] C. Spearman, “General intelligence” objectively determined and measured,” *The American Journal of Psychology*, vol. 15, 1961.