Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **4**, November **2025**

Optimization of Knowledge Graph Relational Reasoning Algorithm Based on Graph Neural Networks

Hui-Yan Zhao*

Zhengzhou Shengda University
Zhengzhou 451191, P. R. China
xczhyan@163.com

Zhou-Xiang Zhen

Faculty Information Science and Technology
National University of Malaysia
Selangor 43600, Malaysia
100618@shengda.edu.cn

*Corresponding author: Hui-Yan Zhao
Received April 12, 2024, revised October 25, 2024, accepted May 20, 2025.

Abstract. *Intending to the issue that the existing Knowledge Graph Reasoning (KGR) algorithms make insufficient use of node structure information, which results in low reasoning efficiency, this article suggests an optimized KGR algorithm based on Graph Neural Network (GNN). Firstly, focusing on the issue of fuzzy key neighborhood information in the process of knowledge graph construction, through two graph attention mechanism levels, neighborhood information is differentiated into triple representation, which ensures the integrity of self-information and neighborhood information to the maximum extent. Secondly, relied on the constructed knowledge graph model, a pre-trained language model is adopted to extract reliable semantic information from the text data and construct a relational association graph, and then Graph Attentional Network (GAN) is adopted to mine the interaction characteristics of the entity nodes in the center of the association graph with the surrounding nodes of different intensities, and finally the pointing information of the nodes is aggregated, so that the network can learn the implied semantic associations between the entities and predict the scores of the triples. Through the knowledge reasoning experiment on the open FB15k dataset, the comparison of relational reasoning accuracy and entity reasoning accuracy with the comparison method has achieved good performance, and successfully verified the efficiency of the optimization algorithm.*
Keywords: Graph neural networks; Knowledge graph; Relational reasoning; Graph attention networks; Pre-trained languages

---

1. **Introduction.** Processing and understanding human knowledge is the ultimate research goal of artificial intelligence. As internet technology develops, human knowledge has spread all over every corner of the internet world at an explosive growth rate. How to efficiently store, manage, search, and utilize this massive knowledge is an urgent issue to be addressed [1]. In this context, the knowledge graph containing the relationship between things opens the real era of “Internet of Everything”, which makes it possible for computers to efficiently use, process, and even understand human knowledge [2]. However, knowledge graph is still faced with many challenges [3, 4, 5]. First, the construction

cost of knowledge graph is high, and most of them are constructed manually or extracted in a semi-automated way. Secondly, there are still many obscured relationships among entities in knowledge graph that are not expressed explicitly, leading to sparse data and a high degree of missing data. Thus, it is urgent to study an automatic reasoning technique to construct knowledge graph efficiently.

Relational reasoning of knowledge graph is a very important research direction. Knowledge graph expresses the knowledge of the world through entities and relationships, and knowledge relationship reasoning is to complete knowledge and discover new knowledge based on the existing information in knowledge graph. Knowledge relation reasoning is a broad and systematic research direction. We believe that through continuous improvement of technology, we can better explore the potential connections in knowledge maps and provide support for the application of artificial intelligence.

1.1. **Related work.** The research on relational reasoning algorithms for knowledge graphs is mainly categorized into traditional reasoning techniques and neural network-based reasoning techniques. Traditional reasoning techniques adopt rule-based and statistic-based approaches. Zhao et al. [6] offered an association rule mining model that supports incomplete knowledge bases, but the reasoning complexity is high, which is unsuitable for large-scale knowledge graph reasoning. Richardson and Domingos [7] suggested a Markov logic network for joint probabilistic reasoning of factual triples in knowledge graphs. Lu et al. [8] introduced the Probabilistic Relational Models (PRMs) of Bayesian networks into the reasoning process to enhance the quality of knowledge graph reasoning. Choudhary et al. [9] suggested a first-order probabilistic linguistic model to enhance the expression of logical rules. Wu et al. [10] adopted a proper pruning strategy to learn rule embeddings from existing triples and new triples inferred from rules to improve the quality of sparse entity embeddings. Heyvaert et al. [11] adopted the idea of acyclic abstract rule templates to optimize the knowledge graph reasoning (KGR) rule generation and evaluation, but the efficiency is not high.

Rule-based models face the issues of low coverage of rules and inefficiency of the reasoning process. Therefore, more and more researchers combine logic rules and neural network techniques, so that the models have both efficient reasoning capabilities. Tiwari et al. [12] used reinforcement learning to solve the task of multi-hop relational reasoning in knowledge graphs. Sang et al. [13] adopted Recurrent Neural Network (RNN) to combine the semantic information of paths of arbitrary length. Wang and Hao [14] adopted an attention-based LSTM network to capture important words on path sequences to improve the performance of relationship classification. Jagvaral et al. [15] adopted an LSTM network to encode the word order information on entity description utterances and then combined it with the attention mechanism embedding model to further learn the semantic information of the triple. Zhang et al. [16] used a convolutional filter to extract 2D reconstructed features of entity and relation embeddings, which enhanced the interaction between entities and relations. However, the above studies only model the internal structure of triples without considering the interactions between triples, which results in insufficient modeling of the knowledge graph.

Recently, the emerging Graph Neural Network (GNN) has been widely used in the field of knowledge graph because of its powerful ability to encode graph structure. Wan et al. [17] adopted the Graph Convolutional Network (GCN) model to process entity neighbor subgraphs, so as to achieve the aggregation of information from the surrounding neighbor entities to the central entity. Zhang et al. [18] adaptively learned the neighborhood structure of the central node by adopting the method of local aggregation. Wu and Zhou [19] used the Graph Attention Network (GAN) to capture the influence of important neighbor context on node update, but the information in key areas was fuzzy. Li et al. [20] adopted two levels of attention, the relational level and the entity level, respectively for reasoning, but they could not accurately represent semantics. The main advantage of GNN in relational reasoning of knowledge graph is that it can better learn the structural information in knowledge graph. The traditional reasoning methods of knowledge graph relations are based on some fixed rules, and it is difficult to capture the complex unstructured relations in knowledge graph. The GNN takes endpoints and edges as inputs and learns the complex interaction of each part of the knowledge graph through the neural network model, so as to understand the internal structure of the knowledge graph more deeply.

1.2. **Contribution.** To deal with the issue that current KGR algorithms invest too many resources in some neighbor regions with weak semantic relevance, which brings interference and noise into the model and leads to low efficiency, this article suggests a KGR algorithm on the ground of GNN optimization.

- To address the issue that the knowledge graph construction model cannot accurately represent the semantics, the graph self-attention mechanism is adopted to enhance the knowledge graph construction model, and the neighborhood information is differentially incorporated into the ternary representation to maximize the completeness of the information. On this basis, a pre-trained language model is adopted to extract reliable semantic information from textual data and construct a relational association graph.
- GAN is adopted to mine the interaction features of different intensities between the entity nodes in the center of the association graph and the surrounding nodes. Finally, the convolutional layer operator is adopted to map the ternary vectors into the obscured-level feature space, and the ternary is processed by the fully connected layer and the activation function to forecast the scores. The experimental outcome indicates that the suggested algorithm improves the Mean Reciprocal Rank (MRR), Mean Rank (MR), and Hits indexes compared with the comparison methods, which illustrates the effectiveness of the GNN-optimized KGR model.

2. Theoretical Analysis.

2.1. **Graphical Attention Network.** GAN introduces the attention mechanism into the graph convolution model and aggregates the neighborhood nodes differently according to the weight of the solution [21]. GNN is a general graph data processing model, and GAN introduces attention mechanism based on graph neural network to improve the flexibility and accuracy of node representation learning. The correlation degree between nodes is indicated in Equation (1).

$$r_{ij} = \beta\,(V\overrightarrow{g_i}, V\overrightarrow{g_j}) = \overrightarrow{b}^{T}\,[V\overrightarrow{g_i} \parallel V\overrightarrow{g_j}] \tag{1}$$

where $\overrightarrow{g_i} \in \mathbb{R}^F$ denotes the feature vector of node i with dimension of size F. $V \in \mathbb{R}^{F\times F}$ is a learnable linear transformation parameter, β denotes the attention mechanism, $\parallel$ denotes the splice of two vectors, and $\overrightarrow{a}^T \in \mathbb{R}^{2RF}$ is a learnable parameter.

The significance of node j to node i can be addressed by the above equation. Finally, the addressed correlation is normalized adopting the softmax function as indicated below:

$$\beta_{ij} = \text{softmax}_j(r_{ij}) = \frac{\exp(r_{ij})}{\sum_{l\in M_i}\exp(r_{il})} \tag{2}$$

By adopting the calculated attention coefficient, the information of the neighborhood nodes can be differentiated and aggregated to complete the convolution operation:

$$\overrightarrow{h_i} = \delta\left(\sum_{j\in M_i} \beta_{ij} V \overrightarrow{g_j}\right) \tag{3}$$

2.2. **Knowledge graph reasoning.** The knowledge graph consists of a series of triples [22], which are represented as $G = \{(h, s, t)\} \subseteq \xi \times \alpha \times \xi$, where ξ and α are the entity (node) set and the relationship (edge) set, respectively. The overall framework is indicated in Figure 1, which consists of four parts: knowledge graph initialization, K-nearest neighbor calculation, three-domain partitioning, and entity and relationship representation learning.

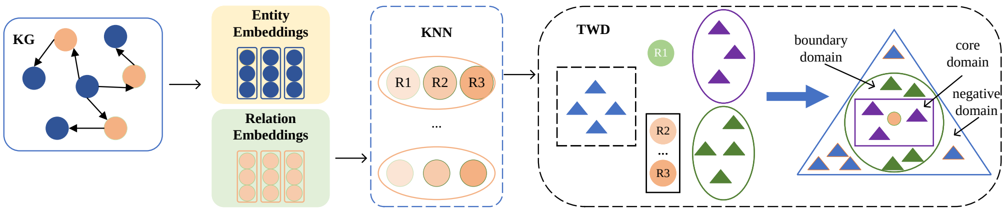

Figure 1. The entire framework of KGR.

First, TransE [23] is adopted to vector-initialize entities and relationships in the knowledge graph. Then, in terms of the idea of K-Nearest Neighbor (KNN) [24] algorithm, the relative nearest neighbor and the corresponding entity set are calculated. The triplet composed of relation and different entity sets is divided into positive domain, boundary domain, and negative domain. Boundary domain is used to represent uncertain knowledge. Finally, the representation learning of the knowledge graph is carried out according to the divided regions, and the embedding learning of the obtained K-nearest neighbor constraint relationship is adopted.

Moreover, the reasoning issue of knowledge graphs can generally be simplified to the following issues:

1. Given the head and tail entities, predict the relationship: $\Gamma_1 = \{g, ?, t\}$, $h, t \in E$.
2. Given an entity and relationship, predict the unknown pseudo-entity: $\Gamma_2 = \{?, s, t\}$ or $\{g, s, ?\}$, $h, t \in E, g \in R$.
3. Given a triple, determine whether it is true or false: $\Gamma_3 = \{g, s, t\}$ true or false, $g, t \in E$, $s \in R$.

3. **Construction of knowledge graph based on GAN.** In view of the ambiguity of the key neighborhood information of the existing knowledge graph model, this article suggests a knowledge graph construction model enhanced by graph self-attention mechanism. The original triplet representation is achieved through TransE training, and the contribution values of different triples are calculated through two levels of graph attention mechanism. The neighborhood information is differentiated into triples to ensure the integrity of its own information and neighborhood information to the maximum extent.

The knowledge graph consists of an SPO (Subject, Predicate, Object) fact triplet, represented as a set $H = (E, R)$ of a directed graph, where E is the set of Subject and Object, and R is the relation. The triplet (e_j, r_i, e_l) represents the existence of an edge F between nodes e_j and e_i in one of the instance graphs H_1. The KGR model adopts the scoring function $F(\cdot)$ to evaluate the efficiency of obtaining triples.

Each graph in the KGR model adopts two embedding matrices as inputs. The entity embedding matrix is represented as $G \in \mathbb{R}^{M_e \times S}$, the matrix i acts as the embedding of entity e_i, M_e is the number of entities, and S is the embedding dimension. Similarly, the matrix $U \in \mathbb{R}^{M_r \times q}$ is represented as a relational embedding. The corresponding embedding matrices are $G' \in \mathbb{R}^{M_e \times S}$ and $U' \in \mathbb{R}^{M_r \times q}$ after each graph is processed by the self-attention level.

To gain entity e_i new embedding $\overrightarrow{g_i}$, it can learn information about other triples in its neighborhood and some of itself, and the attention coefficient β_{jil} between entities to (e_j, e_i) indicates the degree of importance to the central entity. To transform input entity features into higher-level features, the model learns these embeddings by performing linear transformations on local concatenations of entity and relational feature vectors for a particular triplet $t'_{ij} = (e_j, r_i, e_l)$, as shown below:

$$b_{jl} = V_1 \left[\overrightarrow{g_i} \parallel \overrightarrow{u_l} \right] \tag{4}$$

where b_{jl} is the representation of the composite entity vector about the central head entity, and vectors $\overrightarrow{g_i}$ and $\overrightarrow{u_l}$ are the embedding vectors of e_i and r_l, respectively. $V_1 \in \mathbb{R}^{F \times F}$ is a learning weight matrix. $\parallel$ indicates the concatenation operation. In order to obtain more fine-grained feature associations within entities, absolute weights need to be calculated for center-head entity embedding c_{ij} and composite entity embedding c_i.

$$c_{ijl} = LeakyRelu\left(\frac{b_i \cdot b_{jl}}{\sqrt{d_l}}\right) \tag{5}$$

The absolute weights are normalized by softmax to obtain the relative weights of the triples, as indicated below:

$$\beta_{jil}^{e} = \frac{\exp(c_{ijl})}{\sum_{m \in M_i} \sum_{r \in R_m} \exp(c_{imr})} \tag{6}$$

where M_i is the number of first-order neighbors of entity e_i, R_{im} represents the relation set of e_i, and $\exp(\cdot)$ is the exponential function based on the natural constant e. c_{ijl} is the calculated absolute attention coefficient, and β_{ijl}^{e} is the finally obtained relative attention coefficient.

To make the parameters converge more stably during the training process, the model extends the single-headed attention level to multi-headed attention level, so that the Encoder can learn different semantic information in context when learning entity embedding, generate embedding representation for n independent self-attention layers, and then concatenate them to output a vector.

$$\overrightarrow{g_i}' = MulAtt\left(\sum_{j \in M_i} \beta_{ij}^{n} d_{ij}^{n} \overrightarrow{g_{ij}^{n}}\right) \tag{7}$$

where d_{ij}^{n} represents the vector generated by the n-th attention head, and $\overrightarrow{g_i}'$ represents the representation generated after processing by multiple attention layers.

4. **Optimization of Knowledge Graph Relational Reasoning Algorithm Based on Graph Neural Networks.** In view of the insufficient use of node structure information in the process of knowledge reasoning, graph neural network is introduced to enhance the information of nodes in the graph. The whole model is indicated in Figure 2. Firstly, the pre-trained language model is adopted to extract reliable semantic information from the text data, and the relational association graph is constructed. Then, GAN is adopted to capture the interaction features of the central entity node and the surrounding nodes

with different intensities. Finally, the pointing information of the nodes is aggregated, so that the network can learn the implicit semantic association between the entities.

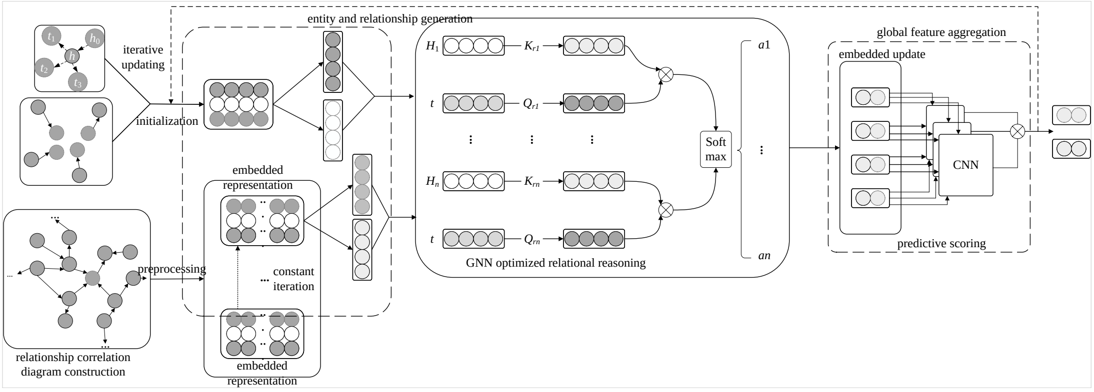

Figure 2. The entire structure of the proposed model.

4.1. **Knowledge graph relationship association graph construction.** Relied on the knowledge graph constructed in the previous section, this article analogizes relationships to labels in multicategorization, and computes the relationship association graph H by mining the co-occurrence patterns of the relationships in the knowledge graph, as well as the semantic similarity between the relationships, as indicated in Figure 3.

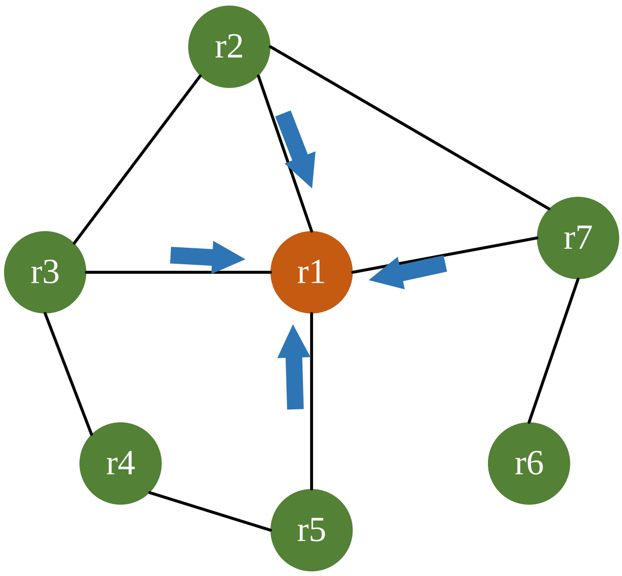

Figure 3. Relational correlation graph.

The amount of co-occurrences between relations is first counted and the co-occurrence matrix is calculated relied on the statistical information.

$$H_{dt}^{1} = \begin{cases} 0, & \text{if } \dfrac{m^{dt}}{m^{d}} < \tau \\ 1, & \text{if } \dfrac{m^{dt}}{m^{d}} \geq \tau \end{cases} \tag{8}$$

where m^d represents the number of triples with relation d, m^{dt} represents the number of triples with relation d and t at the same time, m^{dt}/m^d represents the correlation in the co-occurrence mode, and τ represents the threshold to prevent overfitting of the association matrix. When the co-occurrence frequency of the relation is small, it means that the correlation degree between d and t is small.

However, the amount of long-tail relationships in the knowledge graph is very small, so the co-occurrence matrix cannot reflect the correlation between the relationships well. Therefore, this paper uses the embedded representation trained by Word2Vec [25] to calculate the Euclidean distance between relations, so as to construct the semantic similarity matrix.

$$H_{dt}^2 = \begin{cases} 0, & \text{otherwise} \\ 1, & \text{if } d \in M_w(t) \text{ or } t \in M_w(d) \end{cases} \tag{9}$$

where $M_w(d)$ represents d's nearest neighbor, and then the final relational association matrix H_{dt} is constructed according to the co-occurrence matrix H_{dt}^1 and the semantic similarity matrix H_{dt}^2.

$$H_{dt} = \begin{cases} H_{dt}^2, & \text{otherwise} \\ \dfrac{H_{dt}^1 + H_{dt}^2}{2}, & \text{if } H_{dt}^1 \neq 0 \end{cases} \tag{10}$$

4.2. **Modeling relational reasoning for knowledge graphs based on graph neural networks.** To enhance the issues in the KGR process, relied on the knowledge graph relational association graph constructed above, this article models the knowledge graph relational reasoning task by GAN to reduce the noise of the model.

Suppose that the current input is the features of several nodes, $g = \{\overrightarrow{g_1}, \overrightarrow{g_2}, \dots, \overrightarrow{g_M}\}$, where M represents the number of nodes and N represents the number of characteristics of each node. GAN level needs to learn new node features, $g' = \{\overrightarrow{g_1'}, \overrightarrow{g_2'}, \dots, \overrightarrow{g_M'}\}$, in terms of the obscured characteristics of nodes in the graph and the topological relations between nodes, where the characteristic transformation from g to g' will be calculated by the weight matrix $V \in \mathbb{R}^{N \times N}$.

Specifically, for a pair of nodes (a, b) in the figure, GAN calculates the attention coefficient between (a, b) as indicated below:

$$f_{ab} = \beta(V\overrightarrow{g_a}, V\overrightarrow{g_b}), \quad b \in Z_a \tag{11}$$

where β represents the attention function, Z_a represents the neighbor of node a, and f_{ab} represents the importance of node a to the features of node b.

In general, the model calculates the attention coefficient between the current node and each neighbor, using the softmax function to normalize the selection of all a-nodes, as indicated below:

$$\mu_{ab} = \text{softmax}_b(f_{ab}) = \frac{\exp(\text{LeakyReLu}(f_{ab}))}{\sum_{l \in Z_a} \text{LeakyReLu}(f_{al})} \tag{12}$$

where LeakyReLu$(\cdot)$ is a nonlinear activation function.

After obtaining the normalized attention coefficients, to stabilize the learning process of the attention mechanism, multiple attention heads are involved in the computation, and the multiple attention levels are independent of each other, allowing the model to adaptively choose features in various subspaces. The mapping output of GAN can be computed as follows:

$$\overrightarrow{g_a}' = \delta\left(\frac{1}{L}\sum_{l=1}^{L}\sum_{b \in Z_a} \mu_{ab}^l V^l \overrightarrow{g_b}\right) \tag{13}$$

where δ is the activation function, L is the number of attention heads, and each attention coefficient matrix is computed independently.

Then the information of nodes, relations, and neighbor nodes are simultaneously incorporated into the GAN, and each node in the graph can be defined by its own features, edge features, and related nodes. $\forall a$, which implies potential features on its neighbor Z_a. Also, the relations between the current node connecting to other nodes can be used as an important basis for the relational reasoning of the knowledge graph. The attention value is modified as follows:

$$\mu_{abl} = \text{softmax}_{b,l}(e_{abl}) = \frac{\exp(\text{LeakyReLu}(e_{abl}))}{\sum_{b \in Z_a} \sum_{s \in R_{ab}} \exp(e_{abs})} \tag{14}$$

where Z_a indicates the neighbor of node a, R_{ab} represents the set of relations connecting two entities, $e_{abl} = V\,[\overrightarrow{g_a} \parallel \overrightarrow{g_b} \parallel \overrightarrow{g_l}]$, and $\overrightarrow{g_a}$, $\overrightarrow{g_b}$ are vector representations of nodes. V is a linear transformation matrix, which can be achieved by network training.

The novel entity representation covers the whole context of the local relational subgraph by constantly exchanging information with neighbor and relational nodes.

$$\overrightarrow{g_a^*} = \delta \left(\frac{1}{L} \sum_{l=1}^{L} \sum_{b \in Z_a} \sum_{r \in R_{ab}} \mu_{abl}^{m} f_{abl}^{m} \right) \tag{15}$$

where L represents the number of attention heads. In the final hidden layer of the model, embeddings from multiple heads are averaged to obtain the final embeddings vector of the entity, and the semantic and spatial structure information of each node can be modeled into its representation.

For the purpose of making the network benefit from comparative learning, this paper generates negative samples by randomly replacing the head and tail entities with scrambled corpus, as indicated below:

$$T^- = \left\{(g', s, t) \mid g' \in E \wedge g' \neq g \wedge (g', s, t) \notin T^+\right\} \cup \left\{(g, s, t') \mid t' \in E \wedge t' \neq t \wedge (g, s, t') \notin T^+\right\} \tag{16}$$

where T^+ is the positive triplet sample set, and T^- is the negative triplet sample set.

In this article, the reasoning forecasting task is regarded as a classification task and trained in a supervised way. The loss function of this stage is as follows:

$$\mathcal{L}(g, s, t) = \sum_{t_p \in D_{t_q},\, t_q \in D_{t_p}^-} \max\{d_{t_q} - d_{t_p} + \lambda, 0\} \tag{17}$$

where $d_{t_p} = \|\overrightarrow{h_a} + \overrightarrow{h_r} - \overrightarrow{h_b}\|_{L1/L2}$ and λ are hyperparameters.

4.3. **Global feature aggregation of knowledge graph.** For the goal of integrating characteristics of different levels, this article carried out two-dimensional convolution on the output outcome of GAN networks, fully extracted the global interaction features between nodes, and mapped the features of different aspects between head and tail entity vectors and relational semantic vectors to the same dimension by using convolution kernel of size. To deduce new knowledge from the known information, the convolution result is processed by the fully connected layer and the nonlinear activation function, and finally the triplet knowledge forecasting outcome is achieved. The scoring function definition of the model is indicated below:

$$\Phi(g,s,t) = \left(\sum_{m=1}^{\Omega} \text{LeakyReLu}\left([\overrightarrow{h_a}, \overrightarrow{h_r}, \overrightarrow{h_b}] * \omega^m \right) \right) \cdot W + d \tag{18}$$

where ω stands for convolution kernel and convolution operation, m stands for the number of convolution kernels, W stands for weight, and d stands for deviation term. The loss function is indicated below:

$$\mathcal{L} = \sum_{(g,s,t)\in(T^+\cup T^-)} \log\left(1+\exp\left(l_{(g,s,t)} \cdot \Phi(g,s,t)\right)\right) + \frac{\lambda}{2}\|W\|_2^2 \tag{19}$$

where $l_{(g,s,t)}$ is the label of the positive and negative samples, and W is the weight matrix.

5. Performance testing and analysis.

5.1. Effect analysis of entity reasoning. For the purpose of estimating the performance of the optimized algorithm suggested in this article, simulation experiments were conducted on the publicly available FB15k dataset [26]. FB15k data set is derived from Freebase, the largest general knowledge graph at present, and its data volume is indicated in Table 1. For an Adam model with entity and relationship representation dimensions of {100, 100, 200}, hidden layer size of dimension of 200, output vector dimension of 100, 200, dropout set to 0.2, learning rate size of 0.001, and attenuation coefficient of 0.95, set small-batch training. The size of each batch is set to 50, and softmax and relu are adopted as activation functions. The experiment was based on Windows operating system, Intel (R) Core (TM) i5-13500H CPU @ 12 Hz, 16 GB memory, and Python V3.7.3 simulation environment.

Table 1. Amount of data in FB15k dataset.

Dataset	**entity**	**relation**	**training set**	**validation set**	**test set**
FB15k	16259	257	371589	19575	23947

Four evaluation indexes were adopted to measure the model performance. The triplet with correct reaction ranks among the first, top three, and top ten in the hit ratio of hits@1, hits@3, and hits@10, the average rank of MR (Mean Rank), and the average reciprocal rank of MRR (Mean Rank Reciprocal) were used as comprehensive indexes [27].

Table 2 indicates the comparison outcome of the four models on the FB15k dataset. In order to illustrate the optimization effect of the GNN-KGR model designed in this paper, AR-KGR [18], HR-KGR [20], and TD-KGR [28] reasoning methods were selected for comparison. The black bold font in the table indicates the optimal score on the corresponding indicator.

Through analysis, it is concluded that the hits@1 index of the proposed GNN-KGR reasoning method is about 0.029 lower than that of TD-KGR model, and about 0.057 and 0.025 higher than that of AR-KGR and HR-KGR models, respectively. However, GNN-KGR also has the best performance on hits@3, hits@10, and MR indicators, with an increase of 23.4%, 17.7%, and 10.1% on hits@3, and 15.6%, 9.8%, and 4.6% on hits@10, respectively. MR decreased by 34.8%, 24.6%, and 16.4%.

AR-KGR and HR-KGR models define relation as the rotation of head entity to tail entity in complex vector space, which is suitable for modeling and inferring the relation between entities such as symmetry, anti-symmetry and superposition. The TD-KGR

Table 2. Experimental outcome of entity forecasting on FB15k dataset.

Method	**hits@1**	**hits@3**	**hits@10**	**MR**
AR-KGR	0.386	0.431	0.647	164
HR-KGR	0.418	0.452	0.681	142
TD-KGR	0.472	0.483	0.715	128
GNN-KGR	**0.443**	**0.532**	**0.748**	**107**

model does Tucker decomposition to the binary tensor representing the triplet facts, and uses the shared information to enhance the reasoning effect, but does not further extract the features of the nodes. The GNN-KGR model is also learning the vector relationship between entities and relations in the graph attention module modeling stage, which is more effective for enhancing the interaction between different principal components of entities. On the whole, the GNN-KGR model is effective in extracting node information representation to a certain extent, and has certain reasoning and computing capabilities for matching and associating target entities.

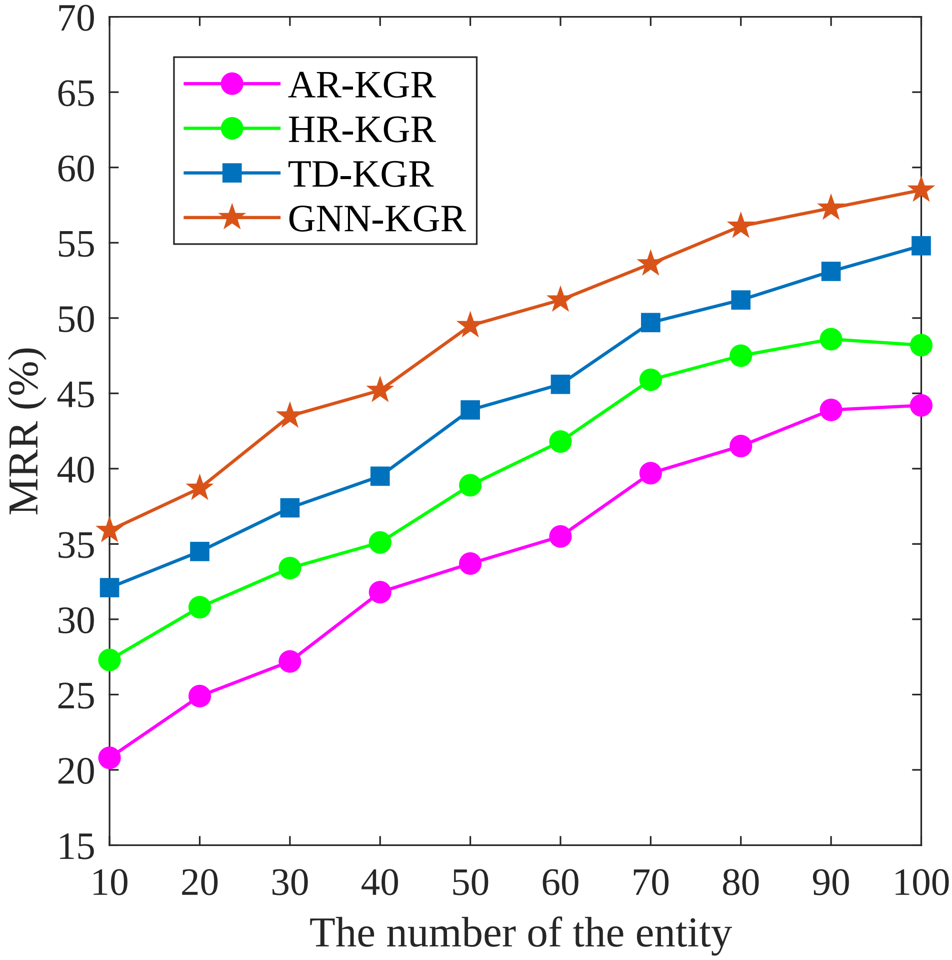

Figure 4. MRR comparison of different models.

The comparative experimental results of the comprehensive index MRR of the four models are indicated in Figure 4. When the amount of entities is 50, the MRR of GNN-KGR method is 49.5%, which is higher than that of AR-KGR, HR-KGR, and TD-KGR methods, which are 15.8%, 10.6%, and 5.6%, respectively. The reason why this method is superior to the comparison is that the GNN-KGR method combines the entity type information and neighborhood information to model the triplet, and assigns different weights to various types of information based on the multi-layer infusing force mechanism, which effectively enhances the effect of knowledge reasoning.

5.2. **Effect analysis of relational reasoning.** Unlike entity prediction, relational reasoning aims to infer the category of relationships between nodes by analyzing the semantic connections between them. Table 3 shows the results of reasoning experiments conducted by GNN-KGR method on different relation types, and it is found that GNN-KGR method performs better on one-to-many and many-to-one relations. This indicates that the suggested GNN-KGR method is suitable for multi-relational knowledge graph and has good applicability.

Table 3. The result of relational reasoning on various types of relation by GNN-KGR method.

Index	**1:1**	**1:M**	**M:1**	**M:M**
hits@1 / %	21.06	32.57	39.28	34.65
hits@3 / %	35.97	45.18	47.39	39.14
hits@10 / %	85.29	95.63	98.27	87.28

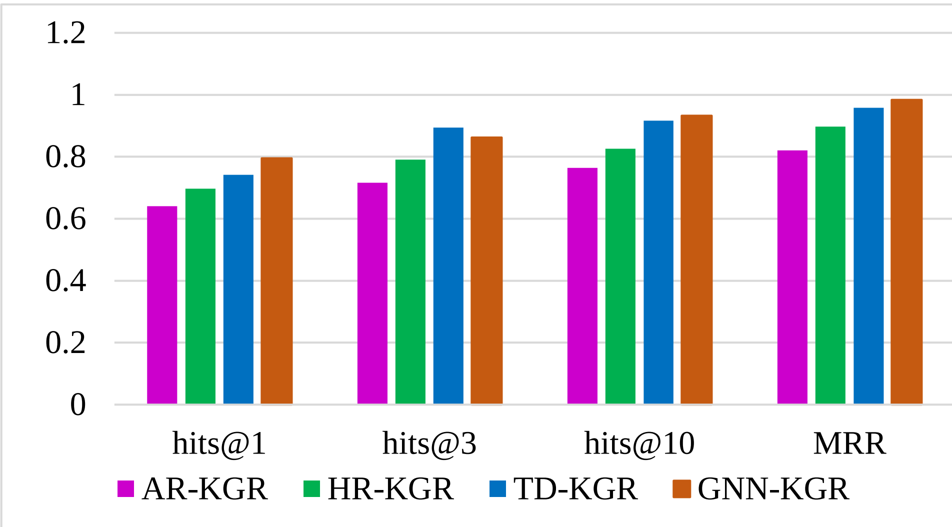

Figure 5. Comparison of relational reasoning performance of different models.

Figure 5 indicates the outcome of the relational reasoning experiments of four models on the FB15k dataset. The GNN-KGR method still performs best, with hits@1, hits@3, hits@10, and MRR reaching 0.795, 0.862, 0.932, and 0.984, respectively. In particular, compared with AR-KGR, HR-KGR, and TD-KGR methods, MRR indexes were increased by 19.85%, 9.7%, and 2.71%, respectively. This shows that the GNN-KGR method can effectively learn semantic information and structural information in the graph, and the rich text information and efficient aggregation features enable the model to capture the implicit relational features of the knowledge graph, which provides more available information for relational reasoning prediction.

6. **Conclusion.** To address the issue of low reasoning efficiency of current KGR algorithms, this article suggests a knowledge graph relationship reasoning algorithm on the ground of GNN optimization. Firstly, focusing on the issue of fuzzy key neighborhood information in the process of knowledge graph construction, the graph self-attention mechanism is adopted to enhance the knowledge graph construction model, calculate the contribution values of different triples, and integrate the neighborhood information into the triples differently to ensure the integrity of information to the maximum extent.

On this basis, the pre-trained language model is used to extract reliable semantic information from the text data, and the relationship association graph is constructed. Then GAN is adopted to capture the interaction characteristics of different strengths between the central entity node of the association graph and the surrounding nodes. Finally, the convolutional neural network is adopted to aggregate the pointing information of the nodes, so that the network can learn the implicit semantic association between the entities. The triples are predicted relied on the information in the association graph structure.

Experimental results show that compared with the comparison method, the proposed algorithm has better performance on MRR, MR, hits@1, hits@3, and hits@10, which indicates the efficiency of GNN-optimized knowledge graph reasoning model.

REFERENCES

[1] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, and Z. Duan, "Knowledge graph completion: A review," *IEEE Access*, vol. 8, pp. 192435–192456, 2020.

[2] J. Yan, C. Wang, W. Cheng, M. Gao, and A. Zhou, "A retrospective of knowledge graphs," *Frontiers of Computer Science*, vol. 12, pp. 55–74, 2018.

[3] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, pp. 112948, 2020.

[4] M. Yani and A. A. Krisnadhi, "Challenges, techniques, and trends of simple knowledge graph question answering: a survey," *Information*, vol. 12, no. 7, pp. 271, 2021.

[5] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[6] F. Zhao, H. Sun, L. Jin, and H. Jin, "Structure-augmented knowledge graph embedding for sparse data with rule learning," *Computer Communications*, vol. 159, pp. 271–278, 2020.

[7] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, pp. 107–136, 2006.

[8] H. Lu, Y. Zidu, W. Yujie, H. Kuang, and Y. Kun, "Link prediction of knowledge graph based on Bayesian network," *Journal of Frontiers of Computer Science & Technology*, vol. 11, no. 5, pp. 742, 2017.

[9] N. Choudhary, N. Rao, S. Katariya, K. Subbian, and C. Reddy, "Probabilistic entity representation model for reasoning over knowledge graphs," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23440–23451, 2021.

[10] C. Wu, Z. Zeng, Y. Yang, M. Chen, X. Peng, and S. Liu, "Task-driven cleaning and pruning of noisy knowledge graph," *Information Sciences*, vol. 646, pp. 119406, 2023.

[11] P. Heyvaert, B. De Meester, A. Dimou, and R. Verborgh, "Rule-driven inconsistency resolution for knowledge graph generation rules," *Semantic Web*, vol. 10, no. 6, pp. 1071–1086, 2019.

[12] P. Tiwari, H. Zhu, and H. M. Pandey, "DAPath: Distance-aware knowledge graph reasoning based on deep reinforcement learning," *Neural Networks*, vol. 135, pp. 1–12, 2021.

[13] L. Sang, M. Xu, S. Qian, and X. Wu, "Knowledge graph enhanced neural collaborative filtering with residual recurrent network," *Neurocomputing*, vol. 454, pp. 417–429, 2021.

[14] Q. Wang and Y. Hao, "ALSTM: An attention-based long short-term memory framework for knowledge base reasoning," *Neurocomputing*, vol. 399, pp. 342–351, 2020.

[15] B. Jagvaral, W.-K. Lee, J.-S. Roh, M.-S. Kim, and Y.-T. Park, "Path-based reasoning approach for knowledge graph completion using CNN-BiLSTM with attention mechanism," *Expert Systems with Applications*, vol. 142, pp. 112960, 2020.

[16] Z. Zhang, Z. Li, H. Liu, and N. N. Xiong, "Multi-scale dynamic convolutional network for knowledge graph embedding," *IEEE Trans. on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2335–2347, 2020.

[17] Y. Wan, Z. Chen, F. Hu, Y. Liu, M. Packianather, and R. Wang, "Exploiting knowledge graph for multi-faceted conceptual modelling using GCN," *Procedia Computer Science*, vol. 200, pp. 1174–1183, 2022.

[18] Z. Zhang, J. Huang, and Q. Tan, "Association rules enhanced knowledge graph attention network," *Knowledge-Based Systems*, vol. 239, pp. 108038, 2022.

[19] Y. Wu and J.-T. Zhou, "A neighborhood-aware graph self-attention mechanism-based pre-training model for knowledge graph reasoning," *Information Sciences*, vol. 647, pp. 119473, 2023.

[20] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, "Learning knowledge graph embedding with heterogeneous relation attention networks," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3961–3973, 2021.

[21] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A spectral convolutional neural network model based on adaptive Fick's law for hyperspectral image classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19–46, 2024.

[22] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a chaotic-based tumbleweed optimization algorithm," *Mathematics*, vol. 11, no. 10, p. 2339, 2023.

[23] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An improved Phasmatodea population evolution algorithm with chaotic maps," *Mathematics*, vol. 11, no. 9, p. 1977, 2023.

[24] S. Dhanabal and S. Chandramathi, "A review of various k-nearest neighbor query processing techniques," *Int. Journal of Computer Applications*, vol. 31, no. 7, pp. 14–22, 2011.

[25] K. W. Church, "Word2Vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[26] Y. Chuanming, Z. Zhengang, and K. Lingge, "Comparing knowledge graph representation models for link prediction," *Data Analysis and Knowledge Discovery*, vol. 5, no. 11, pp. 29–44, 2021.

[27] B. Abu-Salih, "Domain-specific knowledge graphs: A survey," *Journal of Network and Computer Applications*, vol. 185, p. 103076, 2021.

[28] J. Yang, X. Ying, Y. Shi, and B. Xing, "Tensor decompositions for temporal knowledge graph completion with time perspective," *Expert Systems with Applications*, vol. 237, p. 121267, 2024.