electronics

MDPI

Article

Adaptive Graph Neural Network with Incremental Learning Mechanism for Knowledge Graph Reasoning

Junhui Zhang [1,2,3], Hongying Zan [1,*], Shuning Wu [2], Kunli Zhang [1] and Jianwei Huo [2,3]

[1] School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China; zhang_jh@yrec.cn (J.Z.); ieklzhang@zzu.edu.cn (K.Z.)
[2] Yunhe (Henan) Information Technology Co., Ltd., Zhengzhou 450003, China; wsnneversaynever@163.com (S.W.); huojw@yrec.cn (J.H.)
[3] Yellow River Engineering Consulting Co., Ltd., Zhengzhou 450003, China
* Correspondence: iehyzan@zzu.edu.cn

Abstract: Knowledge graphs are extensively utilized in diverse fields such as search engines, recommendation systems, and dialogue systems, and knowledge graph reasoning plays an important role in the aforementioned domains. Graph neural networks demonstrate the capability to effectively capture and process the graph structure inherent in knowledge graphs, leveraging the relationships between nodes and edges to enable efficient reasoning. Current research on graph neural networks relies on predefined propagation paths. The models based on predefined propagation paths overlook the correlation between entities and query relations, limiting adaptability and scalability. In this paper, we propose an adaptive graph neural network with an incremental learning mechanism to search the adaptive propagation path in order to retain promising targets. In detail, the incremental learning mechanism is able to filter out unrelated entities in the propagation path by incorporating the node sampling technique to increase the learning efficiency of the model. In addition, the incremental learning mechanism can select semantically related entities, which promotes the capture of meaningful connections among different nodes in the knowledge graph. At the same time, we apply the parameter initialization module to accelerate the convergence and improve the robustness of the model. Experimental results on benchmark datasets demonstrate that the adaptive graph neural network with the incremental learning mechanism has excellent semantic awareness ability, which surpasses several excellent knowledge graph reasoning models.

Keywords: knowledge graph; knowledge graph reasoning; graph neural network; incremental learning mechanism

check for updates

Citation: Zhang, J.; Zan, H.; Wu, S.; Zhang, K.; Huo, J. Adaptive Graph Neural Network with Incremental Learning Mechanism for Knowledge Graph Reasoning. *Electronics* **2024**, *13*, 2778. https://doi.org/10.3390/electronics13142778

Academic Editor: Dimitris Apostolou

Received: 1 June 2024
Revised: 6 July 2024
Accepted: 8 July 2024
Published: 15 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A knowledge graph is a graph-based representation and organization method for describing entities (such as people, places, and things) and their relationships in the real world [1]. Knowledge graph reasoning refers to the process of utilizing structured information and semantic relationships within a knowledge graph to derive new knowledge or make logical inferences. By leveraging the graph structure and the rich network of interconnected entities and relationships, knowledge graph reasoning enables the exploration of implicit connections and the discovery of hidden patterns [2]. Knowledge graph reasoning has wide-ranging applications across domains such as semantic search [3], question answering systems [4], recommendation systems [5], etc. Knowledge graph reasoning plays a crucial role in enhancing semantic understanding, enabling sophisticated knowledge exploration. Symbolically, the knowledge graph reasoning problem can be defined as a query (subject entity, relation, ?). The objective of knowledge graph reasoning is to seed the target entity for query (subject entity, relation, ?) by learning from the given knowledge graph.

In early research on knowledge graph reasoning, distance-based models like TransE [6] and TransO [7] garnered significant attention for the ability to learn semantic information from knowledge graphs. Distance-based models utilize spatial embeddings in low-dimensional space to represent the relationships among entities in triplets, thereby enabling reasoning. RotatE [8] maps the entities and relations to the complex vector space to enhance the efficiency of reasoning. In addition, RotatE incorporates the self-adversarial negative sampling technique to increase the generalizability of the model. QuatE [9] employs more expressive hypercomplex representations to model entities and relations in knowledge graph reasoning, moving beyond the conventional complex-valued representations. Different from QuatE, the bilinear model ComplEx [10] is designed to predict the target entity by capturing the intrinsic interaction approach of triplets. As the knowledge graph expands, the reasoning methods based on distance for knowledge graphs face significant challenges in handling nonlinear features. Therefore, subsequent research applied deep neural networks to achieve reasoning in large-scale knowledge graphs with nonlinear features. ConvE [11] applies convolution networks to extract interaction information by reshaping the vector into a matrix.

Relying on path-based learning enables capturing local evidence in graphs more effectively, as the approach preserves sequential connections between nodes [12]. The utilization of relational paths marks an initial exploration in capturing both the semantic and local evidence for reasoning purposes [13]. DeepPath [14] employs the reinforcement learning framework to learn multi-hop relational paths, which applies a policy-based agent with continuous states to achieve knowledge graph reasoning. Different from existing embedding-based methods that assume messages are transmitted from nodes to nodes iteratively, the relational message passing of PATHCON [15] only considers edge features to gather neighborhood information. However, the number of paths increases exponentially with the length of the path, which limits the practical application of path-based methods on large-scale knowledge graphs. To solve the scalability issue of path-based methods, AStarNet [16] is designed by searching for important paths instead of using all possible paths for reasoning. AStarNet can reduce time and memory in the process of training and inference.

Subgraphs are inherently more informative than paths in capturing local evidence due to their ability to incorporate multiple nodes and relationships [17,18]. Given the remarkable achievements of graph neural networks in representing data with the graph structure [19], graph neural networks have been employed to capture the subgraph patterns present in knowledge graphs. R-GCN [20] is specifically designed to address the complex nature of knowledge bases, which are characterized by the presence of numerous interrelated relationships. R-GCN excels at effectively handling the highly multi-relational data inherent in realistic knowledge bases. Most graph convolutional networks either focus on representation for nodes or are restricted to graphs with a consistent type of edge, instead of propagating the embeddings of nodes and edges for objectives. KE-GCN [21] combines the power of graph convolutional networks in graph-based belief propagation and the advantage of advanced knowledge embedding methods to address the limitations mentioned above. Specifically, R-GCN conducts propagation across the entire neighborhoods encompassing all entities, which makes it suffer from over-smoothing problems. RED-GNN [22] exhibits superior reasoning capability compared to R-GCN by effectively choosing the range of propagation. Nevertheless, the message propagation of RED-GNN primarily relies on the local neighborhoods of the query entity. As a result, the inclusion of the number of entities increases significantly as the propagation steps increase.

The existing graph neural network-based approaches overlook the semantic correlation between the local neighborhoods and the query relation. From an intuitive perspective, the pertinent local information is contingent upon both the query entity and the query relation. Furthermore, the existing graph neural network-based methods unavoidably incorporate numerous unrelated entities and facts. The unrelated entities and facts intensify the complexity of learning and escalate the computational overhead, particularly in the

case of large-scale knowledge graphs. To alleviate these issues, we propose the adaptive graph neural network with incremental learning mechanism (AGNN-ILM) for knowledge graph reasoning. AGNN-ILM can reduce the number of involved entities while retaining the relevant ones in the propagation path, which is achieved by applying the incremental learning mechanism that incorporates the node sampling mechanism. Node sampling mechanisms have been widely adopted in knowledge graph reasoning tasks based on graph neural networks [23]; they are effective in improving the scalability of the models. The incremental learning mechanisms integrate node sampling techniques and normalization methods to enhance the robustness and scalability of the model. Capturing the intricate semantic dependency between entities in the propagation path and the query relation proves challenging for simplistic heuristics. The incremental learning mechanism employs a semantic-aware distribution to dynamically select local neighborhoods that are relevant to the query relation. In detail, AGNN-ILM initially generates a pool of candidate entities derived from neighboring entities, and subsequently samples from this pool for the next propagation step. At the same time, the incremental learning mechanism samples several entities from the candidate set without replacement, effectively managing the number of entities involved in the propagation process. The incremental learning mechanism exhibits superior efficiency in terms of entities compared to node-wise sampling methods, because the node-wise sampling methods experience an exponential increase in the number of entities involved with each propagation step. In addition, the incremental learning mechanism preserves the layer-wise connection, which improves the sample efficiency. The parameter initialization technique helps the model learn feature representations by setting the distribution of the neural network parameters, thereby improving the generalization capability of the model [24]. Graph neural networks usually consist of multiple layers, the process of gradient propagation may encounter the challenge of gradient vanishing or gradient exploding, further leading to unstable training and hindering the convergence of the model. To enhance the training stability and robustness of graph neural network models, AGNN-ILM incorporates the parameter initialization technique and the layer normalization technique. By introducing the incremental learning mechanism and the parameter initialization module into AGNN-ILM, AGNN-ILM can attain superior stability and scalability.

We evaluate AGNN-ILM on four representative knowledge graph datasets, and the experiment results illustrate that AGNN-ILM can reach state-of-the-art performance on datasets of different scales. In summary, the contributions of AGNN-ILM are concluded as follows:

- We propose an AGNN-ILM algorithm that applies an incremental learning mechanism, which possesses superior semantic awareness and can learn the propagation path adaptively.
- AGNN-ILM incorporates a parameter initialization module to learn the data features of the knowledge graph more quickly, which not only enhances the stability of the training process but also accelerates the convergence of the model.
- The experimental results demonstrate that AGNN-ILM outperforms powerful path-based models and graph neural network-based models on four typical datasets in most cases, with superior stability and scalability.

2. Related Work

A knowledge graph can be represented as $\mathcal{K} = \{\mathcal{V}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{V}$, $\mathcal{R}$, and $\mathcal{F} = \{(e_s, r, e_o) | e_s, e_o \in \mathcal{V}, r \in \mathcal{R}\}$ represent the sets of entities, relations, and fact triples, respectively. Consider the query entity denoted as e_q, the query relation as r_q, and the answer entity as e_a. In the given query $(e_q, r_q, ?)$, the objective of the reasoning knowledge graph task is to predict the unknown answer entity e_a. The key to knowledge graph reasoning lies in capturing local evidence, which refers to the entities and relationships surrounding the query triple (e_q, r_q, e_a). The query triple (e_q, r_q, e_a) is considered a missing link, and the local evidence does not include this specific triplet. Relational paths and subgraphs have

been applied to explore local evidence in knowledge graph reasoning. In this section, the related work on knowledge graph reasoning about path-based methods and graph neural network-based methods are summarized.

2.1. Path-Based Models

A relational path is a grouping of triples that are interconnected in a sequential manner. Path-based methods aim to predict the triple (e_q, r_q, e_a) using a set of relational paths as local evidence. Logic rules are critical to learning hidden information in knowledge graphs, as the rules provide understandable explanations for predictions. At the same time, logic rules possess the capability to generalize to other tasks, which makes the rules indispensable in the process of knowledge graph reasoning. Learning probabilistic first-order logical rules for knowledge base reasoning is challenging, as it involves simultaneously learning parameters in a continuous space and structures in a discrete space. To tackle this challenge, a solution called Neural LP [25] is proposed. Neural LP provides an end-to-end differentiable model that integrates the learning of parameters and structures in first-order logical rules. Similar to Neural LP, DRUM [26] is designed for extracting first-order logical rules from knowledge graphs. Simultaneously, DRUM utilizes bidirectional recurrent neural networks to enhance the exchange of valuable information across different tasks.

Given the impracticality of random walks in scenarios where the destination is unknown, MINERVA [27] learns to traverse the graph based on the input query to facilitate the discovery of predictive paths. Neural LP encounters challenges in navigating a vast search space, while MINERVA faces difficulties in achieving effective optimization for knowledge graph reasoning due to sparse rewards. To overcome the limitations, researchers have introduced a probabilistic model known as RNNLogic [28]. RNNLogic regards logic rules as a latent variable. At the same time, RNNLogic trains a rule generator and a reasoning predictor with logic rules. Many scholars have made multiple attempts to automatically learn logical rules from knowledge graphs. Nonetheless, the majority of existing methods heavily depend on observed instances of rules to establish the scoring function for rule evaluation. Consequently, the methods often face challenges in terms of generalization ability and computational efficiency. In contrast to the conventional approach of solely relying on rule instances for rule evaluation, RLogic [29] proposes a scoring model based on predicate representation learning. RLogic is trained using sampled rule instances, providing an alternative methodology for rule evaluation. In addition, RLogic incorporates the deductive nature into rule learning, which is especially essential when a rule does not possess sufficient supporting evidence. In order to further integrate deductive reasoning into the rule learning process, RLogic employs a recursive approach to decompose a large sequential model into smaller atomic models.

2.2. Graph Neural Network-Based Models

Subgraphs possess the capability to preserve richer information due to their ability to accommodate a greater degree of connectivity, in contrast to relational paths. Meanwhile, graph neural networks exhibit strong modeling power when it comes to graph-structured data. A Markov logic network effectively combines the power of first-order logic to leverage domain knowledge while handling uncertainty. Different from the Markov logic network, knowledge graph embedding methods focus on learning entity and relation embeddings to facilitate reasoning. pLogicNet [30] integrates the strengths of both approaches, which defines the joint distribution of each possible triplet using a Markov logic network. Multi-relational graphs are a widely encountered and more comprehensive type of graph, where each edge is labeled and possesses a specific direction. However, existing methods face challenges related to excessive parameterization when addressing multi-relational graphs. CompGCN is a novel graph convolutional framework, proposed in [31], which aims to jointly embed both nodes and relations in a relational graph. CompGCN utilizes a range of entity-relation composition operations, allowing it to effectively handle a large number of relations. DPMPN [32] applies the graph neural network framework to encode

input-agnostic full graph representations. In addition, DPMPN learns input-dependent local collaboration through an attention module. By incorporating a graphical attention mechanism conditioned on input queries, the construction and expansion of subgraphs are dynamically performed in a flexible manner. Through the approach, DPMPN not only constructs the explanations of graph structure but also enables message passing in graph neural networks to scale with the sizes of the graph.

Inspired by traditional path-based methods, a versatile and adaptable representation learning framework is proposed in NBFNet [33]. Additionally, NBFNet can effectively enhance the capacity of the path formulation, leading to improved performance and capabilities. A novel relational structure named a relational directed graph is designed in RED-GNN [22] to capture local evidence of knowledge graphs, which consists of overlapped relational paths. Encoding the relational directed digraphs directly faces challenges in scalability, as it may not efficiently handle larger graph sizes. Additionally, capturing query-dependent information becomes a difficult task within the context of relational-directed digraphs. RED-GNN utilizes dynamic programming to recursively encode multiple relational directed digraphs that share edges. Additionally, RED-GNN integrates a query-dependent attention mechanism to select highly correlated edges. The determination of the semantic relevance of entities to a specific query lacks direct oversight and straightforward metric-based evaluation when designing the sampling algorithm. To address the challenge, AdaProp [34] incorporates a learning-based sampling distribution that can dynamically adjust to different queries. Most inductive knowledge graph completion methods assume that new entities can be added to a knowledge graph, but the methods do not account for the possibility of new relations emerging during inference. This limitation hampers their effectiveness in handling real-world knowledge graphs where both new entities and new relations are introduced. INGRAM [35] addresses the challenge by enabling the generation of embeddings for both new entities and new relations during the inference process. INGRAM conceptualizes a relation graph as a weighted representation that incorporates relations and the corresponding affinity weights. In addition, INGRAM learns aggregating neighboring embeddings to construct relation and entity embeddings with an attention mechanism. C-MPNN [36] defines a relational Weisfeiler–Leman algorithm, and it can be demonstrated that conditional message passing neural networks possess superior expressive power. In addition, the study of C-MPNN provides intriguing findings regarding NBFNet, indicating that the impressive empirical performance of NBFNet can be attributed to expressive power.

3. Model

In this section, we first introduce the motivation for designing AGNN-ILM. Then, the specific structural composition of AGNN-ILM is demonstrated in detail.

3.1. Motivation

Graph neural network-based knowledge graph reasoning methods incorporate a critical design element called propagation paths, which are sequences of entities involved in each step of the propagation process. However, current approaches rely on manually crafted propagation paths, neglecting the relationship between the entities and the queried relation. Furthermore, with an increase in propagation steps, the engagement of entities grows exponentially. To alleviate the issues, AGNN-ILM is designed to learn an adaptive propagation path that effectively filters out irrelevant entities while retaining promising targets. In detail, AGNN-ILM employs the incremental learning mechanism to preserve the previous step entities. In addition, AGNN-ILM samples the candidate entities from newly visited entities to avoid discarding the promising targets. To address the potential training fluctuations arising from the intricate message passing in graph neural networks, the parameter initialization technique is employed in AGNN-ILM. By incorporating the parameter initialization technique along with the incremental learning mechanism, AGNN-ILM achieves superior stability and delivers satisfactory performance.

3.2. Adaptive Graph Neural Network with Incremental Learning Mechanism

The framework of AGNN-ILM is shown in Figure 1. AGNN-ILM initially applies the incremental learning mechanism to acquire neighboring nodes and generates a collection of newly visited nodes, which are subsequently regarded as candidate nodes. Subsequently, the incremental learning mechanism computes the independent query messages by leveraging the corresponding edges. Afterward, candidate node information is aggregated, and the nodes are sampled to generate hidden representations. The predicted target entity is then identified by evaluating the highest score derived from the hidden representations. AGNN-ILM consists of n layers of graph neural networks, and the propagation path of AGNN-ILM is adaptively adjusted instead of fixed before the message propagation. The model structure of AGNN-ILM is shown in Figure 2. In detail, AGNN-ILM obtains the neighboring entities in the nth propagation step by

$$\mathcal{N}(\mathcal{V}^{n-1}_{e_q,r_q}) = \cup_{e \in \mathcal{V}^{n-1}_{e_q,r_q}} \mathcal{N}(e), \tag{1}$$

where $\mathcal{V}^{n-1}_{e_q,r_q}$ represents the entities in the $(n-1)$-th step. At the same time, the incremental learning mechanism generates the newly visited entities as candidates by

$$\overline{V}^{n}_{e_q,r_q} = \mathcal{N}(\mathcal{V}^{n-1}_{e_q,r_q}) \setminus \mathcal{V}^{n-1}_{e_q,r_q}. \tag{2}$$

Then, the edges related to (e_q, r_q) are calculated by

$$\mathcal{E}^n = \{(e_s, r, e_o) | e_s \in \mathcal{V}^{n-1}_{e_q,r_q}, e_o \in \mathcal{N}(\mathcal{V}^{n-1}_{e_q,r_q})\}. \tag{3}$$

For $(e_s, r, e_o) \in \mathcal{E}^n$, r is processed with the embedding layer to output r_e by

$$r_e = Embedding(r). \tag{4}$$

r_q is also processed by the embedding layer to output r_{qe}. Using h^{n-1} denotes the hidden representation at the $(n-1)$-th propagation step, which is initialized with zero. h^{n-1} is updated with the propagation step. The attention of query-dependent messages about $\mathcal{E}^n$ is calculated by

$$W^m_a = W_a(RELU(W_{s_a} h^{n-1}[e_s] + W_{r_a} r_e + W_{qr_a} r_{qe})), \tag{5}$$

where W_a, W_{s_a}, W_{r_a}, W_{qr_a} are the parameters, and ReLU [37] is applied to make the model learn complex patterns and features. After that, the query-dependent messages are computed by

$$m^n_{(e_s,r,e_o)} = W^m_a(h^{n-1}[e_s] + r_e)). \tag{6}$$

The messages propagate to entities in $\mathcal{N}(\mathcal{V}^{n-1}_{e_q,r_q})$ and the entity representations of e_o are obtained by

$$h^n[e_o] = AGG(m^n_{(e_S,r,e_o)}), \tag{7}$$

where AGG represents the aggregation function. The model learns the semantic-aware distribution by

$$p^n(e) := \exp(f(h^n[e_o]; \theta^n)/\tau) \Big/ \sum_{e'_o \in \overline{V}^n_{e_q,r_q}} \exp\Big(f(h^n[e'_o]^n; \theta^n)/\tau\Big), \tag{8}$$

where $f(h^n[e_o]; \theta^n)$ represents the linear mapping function, θ^n represents the parameters of AGNN-ILM, and the temperature value τ is set to 1. According to the semantic-aware distribution, the logits of e_o are computed by

$$G(e_o) = g(h^n[e_o]), \tag{9}$$

where g represents the gumbel softmax activation function [38]. The sampled entities are attained by

$$\widetilde{\mathcal{V}}^{n}_{e_q,rq} = \arg\operatorname{top}_K G(e_o). \tag{10}$$

The propagation path is updated by

$$\mathcal{V}^{n}_{e_q,r_q} = \mathcal{V}^{n-1}_{e_q,r_q} \cup \widetilde{\mathcal{V}}^{n}_{e_q,r_q}. \tag{11}$$

After propagating for N steps, the involved entities' representations $h^n[e_o]$ are obtained. The entity with the highest score in $\mathcal{V}^{n}_{e_q,r_q}$ is regarded as the predicted target. The incremental learning mechanism samples relevant entities based on the semantic perception distribution, which can effectively retain the target entities. At the same time, the parameter initialization techniques are applied to enhance the stability and robustness of the model. In detail, the parameter initialization techniques enable the network to maintain stable gradient propagation during training by setting the initial value distribution of weight parameters, thereby accelerating the convergence of the model. The parameter initialization techniques avoid reducing or magnifying the scale of input signals exponentially by

$$\frac{1}{2} n_l \mathrm{Var}[w_l] = 1, \tag{12}$$

where n_l represents the input dimensionality of the l-th layer. The weights of the l-th layer w_l are initialized by

$$w_l \sim \mathcal{N}(0, 2/n_l), \tag{13}$$

where $\mathcal{N}(0, \frac{2}{n_l})$ denotes a Gaussian distribution with mean 0 and standard deviation $\sqrt{\frac{2}{n_l}}$. The loss function L adopts the binary cross-entropy loss, which is calculated by

$$L = -\sum_{e_o \in V^{n}_{eq,rq}} y_{e_o} log(l_{e_o}) + (1 - y_{e_o}) log(1 - l_{e_o}), \tag{14}$$

where $y_{e_o} = 1$ if the predicted target e_o is equal to e_a, otherwise 0. l_{e_o} represents the likelihood score for entity e_o that indicates the rationality of e_o as the target entity. AGNN-ILM is the graph neural network-based model, thus the computational complexity of AGNN-ILM is similar to that of graph neural networks. As the number of network layers increases, the time complexity increases linearly. The computational complexity of AGNN-ILM grows linearly with the scale of the knowledge graph rather than exponentially by incorporating the incremental learning mechanism, which endows AGNN-ILM with excellent scalability.

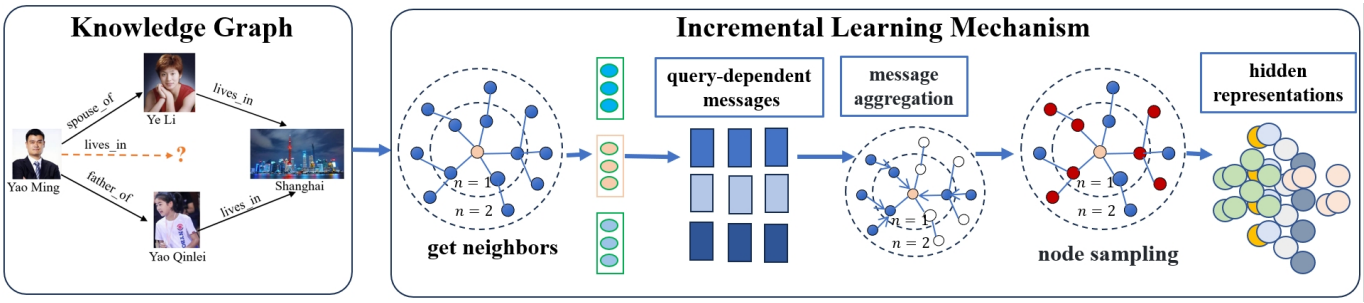


Figure 1. The framework of AGNN-ILM.

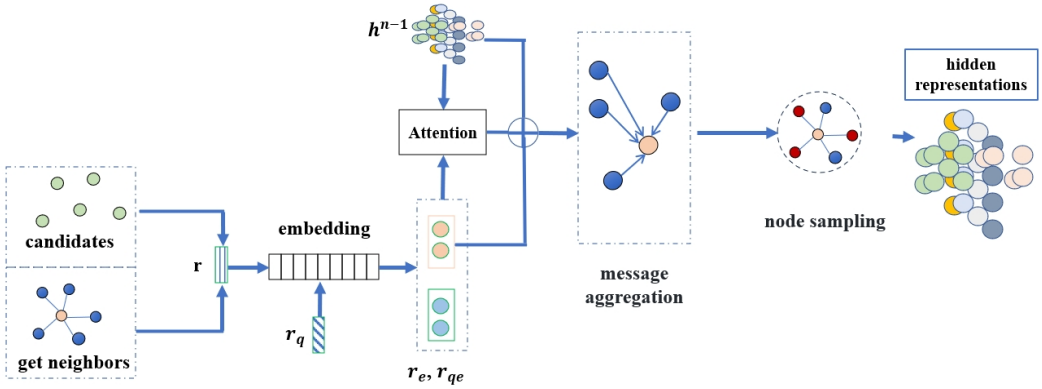


Figure 2. The model structure of AGNN-ILM.

4. Experiments

This section begins with an introduction to the metrics used for evaluating algorithm performance. Next, the datasets utilized in the experiment are described, followed by a description of the experimental setup. Finally, an analysis of the experimental results is presented.

4.1. Evaluation Indicator

To assess the outcomes in the ranking task, we employ two commonly used metrics: mean reciprocal rank (MRR) and Hits@N ($N \in \{1, 10\}$). The metrics are widely recognized for evaluating the performance of algorithms in the knowledge graph reasoning task. MRR is calculated by

$$MRR = \frac{1}{n_t} \cdot \sum_{i}^{n_t} \frac{1}{\text{rank}_i}, \tag{15}$$

where n_t represents the number of triples in the test set; rank_i refers to the ranking of link prediction for the i-th triple. Hits@N (H@N) is computed by

$$Hits@N = \frac{1}{n_t} \cdot \sum_{i}^{n_t} f(rank_i), \quad f(rank_i) = \begin{cases} 1, x \leq rk \\ 0, x > rk \end{cases}. \tag{16}$$

If $\text{rank}_i \leq rk$ is true, the value of function $f(rank_i)$ is 1. The higher the values of MRR and Hits@N, the better the performance of the algorithm.

4.2. Datasets

AGNN-ILM is assessed on four extensively employed benchmark datasets, namely, Family [39], UMLS [39], WN18RR [11], and NELL-995 [14]. The detailed information of the datasets is shown in Table 1. UMLS serves as a comprehensive repository that consolidates a wide range of health and biomedical terminologies and standards. Family is utilized to investigate knowledge representation and reasoning pertaining to family relationships. The UMLS and Family datasets contain relatively few entities, making them suitable choices to represent small-scale datasets. WN18RR, a subset of WN18 [6], is an extended version obtained by reversing and randomly sampling relations from the WN18 dataset. WN18RR is recognized for its effectiveness in evaluating model performance, making it an ideal representative of medium-scale datasets. NELL-995 offers hierarchical information on entities and relationships, encompassing a wide range of domains. With a significantly larger number of entities and relationships compared to the other datasets used in this paper, NELL-995 serves as the representative of large-scale datasets.

Table 1. Statistics of the knowledge graph datasets.

Dataset	Entity	Relation	$\lvert\mathcal{E}\rvert$	Train	Valid	Test
Family [39]	3007	12	23,483	5868	2038	2835
UMLS [39]	135	46	5327	1321	569	633
WN18RR [11]	40,903	11	86,835	21,705	3034	3134
NELL-995 [14]	74,536	200	112,258	37,420	543	2818

4.3. Experiment Setup

The parameter settings of the model are as shown in Table 2. We employ varied values and learning rates tailored to different benchmark datasets. The number of layers in a graph neural network affects the complexity and performance of the model, thus requiring appropriate layer configuration. Due to the small scale of the UMLS and Family datasets, the number of sampled nodes is smaller compared to WN18RR and NELL-995. At the same time, the dimension of the hidden layer is positively correlated with the size of the datasets. The experiments are conducted using the PyTorch framework and executed on a single NVIDIA RTX 6000 GPU with 48 GB of memory.

Table 2. Detailed settings of AGNN-ILM. *lr* represents the learning rate of AGNN-ILM. *bs* represents the batch size. *n* represents the number of graph neural network layers in AGNN-ILM. *k* represents the number of sampled nodes. *epoch* represents the total number of training iterations. *d* represents the dimensionality of the hidden layer of AGNN-ILM.

Dataset	*lr*	*bs*	*n*	*k*	*epoch*	*d*
Family	0.0036	20	8	100	300	48
UMLS	0.0012	10	6	100	300	64
WN18RR	0.0030	50	8	1000	300	64
NELL-995	0.0011	10	6	2000	100	128

4.4. Experiment Results

We compared the performance of AGNN-ILM with several benchmark models on the Family, UMLS, WN18RR, and NELL-995 datasets. The benchmark models contain distance-based models (i.e., RatatE, QuatE), path-based methods (i.e., MINERVA, Neural LP), and graph neural network-based models (i.e., CompGCN, DPMPN, RED-GNN). The results of all models on the Family and UMLS datasets are shown in Table 3.

Table 3. The experimental results of Family and UMLS. Best performances are shown in bold, and the underlined values indicate the second best. The results of AdaProp are taken from [34] and the other results are taken from [22].

Comparison	Family			UMLS		
	MRR	H@1	H@10	MRR	H@1	H@10
RotatE [8]	0.921	0.866	0.988	0.925	0.863	0.993
QuatE [9]	0.941	0.896	0.991	0.944	0.905	0.993
MINERVA [27]	0.885	0.825	0.961	0.825	0.728	0.968
Neural LP [25]	0.924	0.871	<u>0.994</u>	0.745	0.627	0.918
CompGCN [31]	0.933	0.883	0.991	0.927	0.867	0.994
DPMPN [32]	0.981	0.974	0.981	0.930	0.899	0.982
RED-GNN [22]	**0.992**	<u>0.988</u>	**0.997**	0.964	0.946	0.990
AdaProp [34]	0.988	0.986	0.990	<u>0.969</u>	<u>0.956</u>	<u>0.995</u>
AGNN-ILM (ours)	<u>0.990</u>	**0.989**	0.991	**0.988**	**0.983**	**0.998**

It can be noted that the graph neural network-based models are superior to the distance-based models and path-based models generally. According to the results of Family,

the overall performance of AGNN-ILM is superior to all other algorithms except RED-GNN. AGNN-ILM achieved the best performance in terms of H@1. From the results of UMLS, it can be noted that the performance of AdaProp is inferior to AGNN-ILM while outperforming the other algorithms. AGNN-ILM demonstrates overwhelming superiority compared with the other benchmark models. AGNN-ILM achieves the best performance in all evaluation indicators on the UMLS dataset.

The experimental results on the WN18RR and NELL-995 datasets are shown in Table 4. Based on the experimental results, it can be observed that AGNN-ILM outperforms other models significantly on medium-scale and large-scale datasets. Compared with RED-GNN on WN18RR, AGNN-ILM improves the MRR from 0.533 to 0.568, H@1 from 0.485 to 0.519, and H@10 from 0.624 to 0.666. Although AGNN-ILM has a lower H@10 value compared to AdaProp, AGNN-ILM surpasses the state-of-the-art model AdaProp in terms of MRR and H@1. On the NELL-995 dataset, AGNN-ILM achieves a performance improvement of 2.5% in terms of MRR, 3.9% in terms of H@1, and 0.8% in terms of H@10 compared to RED-GNN. The overall performance of AGNN-ILM is inferior to AdaProp on NELL-995, which demonstrates that AdaProp has excellent reasoning capabilities when dealing with knowledge graphs with massive entities.

In summary, AGNN-ILM achieves excellent performance across the four datasets. The experimental results on datasets of various scales demonstrate the enormous potential of AGNN-ILM in knowledge graph reasoning. Meanwhile, it is demonstrated that AGNN-ILM can be the state-of-the-art graph neural network-based model in large knowledge graphs. Furthermore, the results indicate that AGNN-ILM has the potential to become the state-of-the-art graph neural network-based model for large knowledge graphs.

Table 4. The experimental results on WN18RR and NELL-995. Best performances are shown in bold, and the underlined values indicate the second best. '-' means unavailable results. The results of AdaProp are taken from [34] and the other results are taken from [22].

Comparison	WN18RR			NELL-995		
	MRR	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
RotatE [8]	0.477	0.428	0.571	0.508	0.448	0.608
QuatE [9]	0.480	0.440	0.551	0.533	0.466	0.643
MINERVA [27]	0.448	0.413	0.513	0.513	0.413	0.637
Neural LP [25]	0.435	0.371	0.566	-	-	-
CompGCN [31]	0.479	0.443	0.546	-	-	-
DPMPN [32]	0.482	0.444	0.558	0.513	0.452	0.615
RED-GNN [22]	0.533	0.485	0.624	0.543	0.476	<u>0.651</u>
AdaProp [34]	<u>0.562</u>	<u>0.499</u>	**0.671**	**0.554**	**0.493**	**0.655**
AGNN-ILM (ours)	**0.568**	**0.519**	<u>0.666</u>	<u>0.553</u>	<u>0.490</u>	**0.655**

4.5. Ablation Study

In this subsection, we conducted a set of ablation experiments to verify the roles of the parameter initialization module and the incremental learning module.

4.5.1. Effect of the Parameter Initialization Module

AGNN-ILM integrates the parameter initialization module to enhance model stability and robustness. The parameter initialization model works by establishing an appropriate initial distribution for the weight parameters, which ensures stable gradient propagation throughout the training process. To verify the stability of the parameter initialization module, we compared the performance of AGNN-ILM and AGNN-ILM without the parameter initialization module (AGNN-ILM/PI) across four datasets. The experimental results are presented in Table 5. As can be seen from Table 5, AGNN-ILM outperforms AGNN-ILM/PI in knowledge graph datasets of different scales, demonstrating that the parameter initialization module can effectively enhance the performance of the model.

The training curves of AGNN-ILM and AGNN-ILM/PI on Family are shown in Figure 3. Figure 3 reveals that AGNN-ILM/PI exhibits a declining trend in performance after 100 epochs, in contrast to AGNN-ILM. The experimental results on Family demonstrate that the parameter initialization module effectively enhances the stability of the model. Figure 4 indicates that on the UMLS dataset, AGNN-ILM outperforms AGNN-ILM/PI significantly in terms of MRR and H@1. Additionally, AGNN-ILM demonstrates faster convergence compared to AGNN-ILM/PI, highlighting the role of the parameter initialization module in accelerating the convergence speed of the model. Figure 5 exhibits significant fluctuations in the training curve of AGNN-ILM/PI on WN18RR. On the other hand, the training curve of AGNN-ILM exhibits notable robustness, showcasing the effectiveness of the parameter initialization module in enhancing model stability. Figure 6 demonstrates that on NELL-995, AGNN-ILM outperforms AGNN-ILM/PI in terms of both performance and stability. In summary, the parameter initialization module effectively enhances the stability, robustness, and performance of the model across knowledge graph datasets of different scales.

Table 5. The ablation experimental results for the parameter initialization module on Family, UMLS, WN18RR, and NELL-995. Best performances are shown in bold.

Comparison	Family			UMLS		
	MRR	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
AGNN-ILM/PI	0.988	0.988	0.990	0.974	0.963	0.995
AGNN-ILM	**0.990**	**0.989**	**0.991**	**0.988**	**0.983**	**0.998**
Comparison	**WN18RR**			**NELL-995**		
	MRR	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
AGNN-ILM/PI	0.560	0.509	0.660	0.549	0.487	0.653
AGNN-ILM	**0.568**	**0.519**	**0.666**	**0.553**	**0.490**	**0.655**

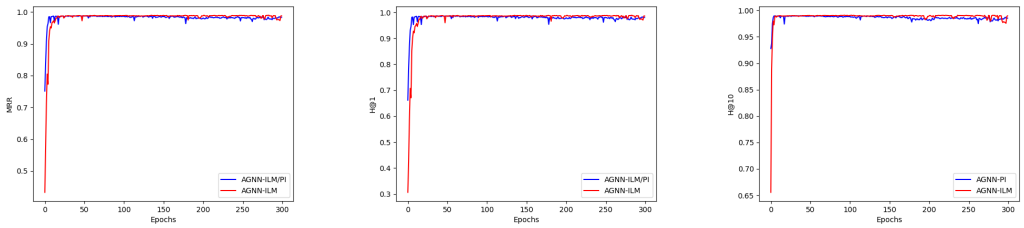


Figure 3. The ablation experimental results for the parameter initialization module on Family dataset.

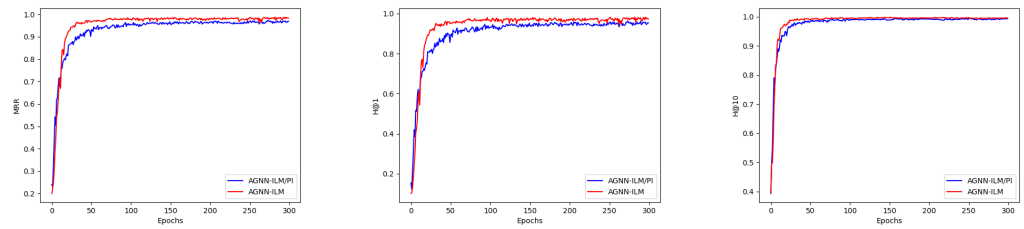


Figure 4. The ablation experimental results for the parameter initialization module on UMLS dataset.

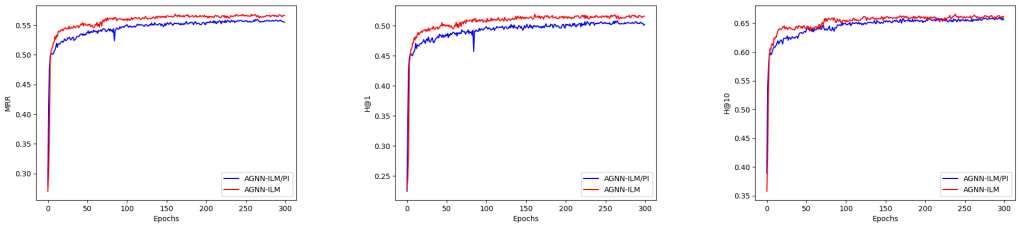


Figure 5. The ablation experimental results for the parameter initialization module on WN18RR dataset.

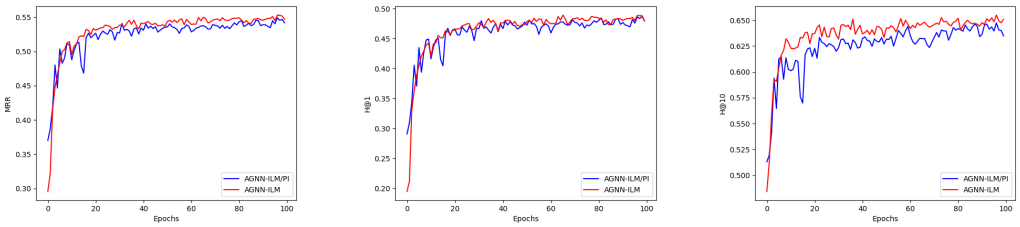


Figure 6. The ablation experimental results for the parameter initialization module on NELL-995 dataset.

4.5.2. Effect of the Incremental Learning Module

To validate the impact of the layer normalization technique in the incremental learning module, we conducted a performance comparison between AGNN-ILM and AGNN-ILM without the layer normalization technique (AGNN-ILM/LN). As shown in Table 6, AGNN-ILM and AGNN-ILM/LN achieved the same performance on the UMLS dataset. It can be noted that AGNN-ILM demonstrated more stable performance compared with AGNN-ILM/LN during the training process in Figure 7, indicating the role of the layer normalization technique in improving the stability of the model. As can be seen in Figure 8, AGNN-ILM/LN exhibited slower convergence speed and inferior performance compared to AGNN-ILM, demonstrating that the layer normalization technique can effectively improve the performance of the model.

Table 6. The ablation experimental results for the layer normalization technique in the incremental learning mechanism on Family and UMLS. Best performances are shown in bold.

Comparison	Family			UMLS		
	MRR	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
AGNN-ILM/LN	**0.990**	**0.989**	**0.991**	0.976	0.965	0.996
AGNN-ILM	**0.990**	**0.989**	**0.991**	**0.984**	**0.976**	**0.998**

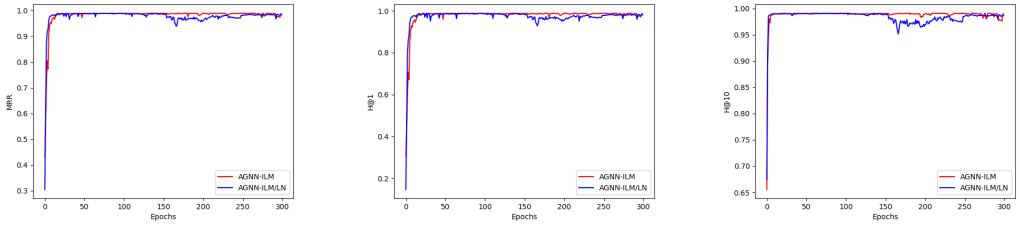


Figure 7. The ablation experimental results for the layer normalization technique in the incremental learning mechanism on Family dataset.

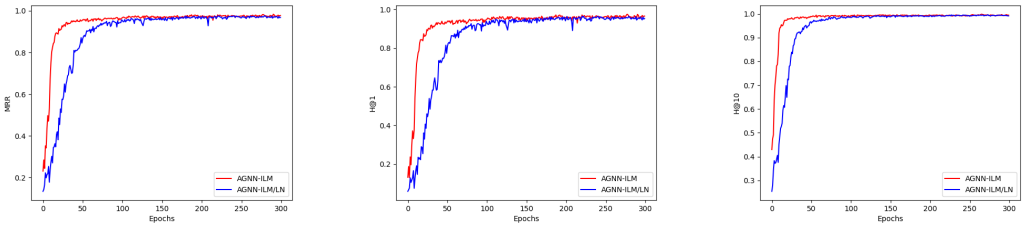


Figure 8. The ablation experimental results for the layer normalization technique in the incremental learning mechanism on UMLS dataset.

We compare the performance of AGNN-ILM and AGNN without the incremental learning module (AGNN) to validate the role of the incremental learning module. Table 7 presents the experimental results of AGNN-ILM and AGNN on knowledge graph datasets of various scales. As can be clearly seen in Table 7, AGNN-ILM outperforms AGNN on all metrics on both the UMLS and Family datasets. In WN18RR and NELL-995, AGNN-ILM lags behind AGNN in terms of H@10. We attribute the decline in model performance to the reduced probability of sampling the target entities as the knowledge graph scale increases. Although AGNN-ILM performs relatively worse than AGNN in terms of H@10, AGNN-ILM exhibits superior overall performance compared to AGNN. In summary, the incremental learning module effectively enhances the semantic awareness of the model, leading to notable performance improvements.

Table 7. The ablation experimental results on the incremental learning mechanism on Family, UMLS, WN18RR, and NELL-995. Best performances are shown in bold.

Comparison	**Family**			**UMLS**		
	MRR	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
AGNN	0.988	0.987	0.990	0.973	0.958	0.997
AGNN-ILM	**0.990**	**0.989**	**0.991**	**0.988**	**0.983**	**0.998**
Comparison	**WN18RR**			**NELL-995**		
	MRR	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
AGNN	**0.568**	0.514	**0.673**	0.545	0.479	**0.657**
AGNN-ILM	**0.568**	**0.519**	0.666	**0.553**	**0.490**	0.655

To investigate the impact of the number of sampling nodes in the incremental learning module on model performance, we conducted an ablation experiment on the UMLS dataset. The experimental results are shown in Table 8. In the scenario where AGNN-ILM incorporates a five-layer graph neural network, the model performance initially shows a decline but subsequently improves with an increase in the number of sampled nodes. In the scenario where the AGNN-ILM consists of a six-layer graph neural network, the performance of AGNN-ILM improves as the number of sampling nodes increases, reaching the peak when the number of sampling nodes reaches 100. When the number of sampling nodes is set to 100, AGNN-ILM achieves optimal performance regardless of whether n is 5 or 6. The overall performance of AGNN-ILM is inferior to that of $k = 100$ when k is set to 120, which demonstrates that increasing the number of sampling nodes does not necessarily lead to performance improvement. Figure 9 presents the training curves of the model with different numbers of sampled nodes when n is set to 5 or 6. From Figure 9, we can observe that the performance of the model is not significantly affected by the different numbers of sampled nodes when $n = 5$. The number of sampled nodes significantly affects the convergence speed of the model when $n = 6$. It can be noted that the model exhibits the fastest convergence speed when $k = 100$. Both an excessive or insufficient number of sampled nodes can impact the convergence speed of the model. Therefore, establishing an appropriate number of sampling nodes is crucial for guaranteeing model performance.

Table 8. Ablation study on the number of sampled nodes in the incremental learning module in UMLS. Best performances are shown in bold, and the underlined values indicate the second best.

Comparison	n = 5			n = 6		
	MRR	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
k = 50	**0.984**	**0.976**	0.996	0.976	0.964	<u>0.996</u>
k = 60	<u>0.982</u>	0.973	<u>0.997</u>	0.978	0.968	<u>0.996</u>
k = 80	<u>0.982</u>	<u>0.975</u>	0.996	0.981	0.971	**0.998**
k = 100	**0.984**	**0.976**	**0.998**	**0.988**	**0.983**	**0.998**
k = 120	**0.984**	**0.976**	<u>0.997</u>	0.985	0.978	**0.998**

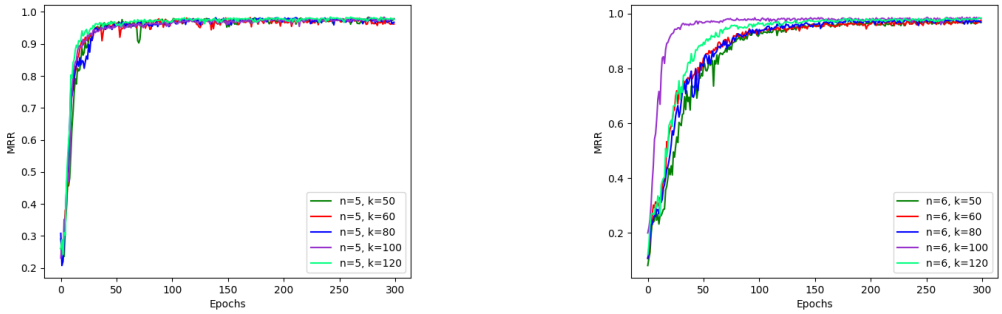


Figure 9. Experimental results on the number of sampled nodes in the incremental learning module on the performance of AGNN-ILM on UMLS dataset.

4.5.3. Effect of the Number of Graph Neural Network Layers

To verify the impact of the number of graph neural network layers on the performance of AGNN-ILM, we conducted ablation studies on the UMLS dataset. The statistical results are presented in Table 9. It can be observed that increasing the number of network layers does not necessarily lead to improved performance of AGNN-ILM. AGNN-ILM achieves the best performance when the number of layers of the graph neural network is six.

Table 9. Experimental results on the effect of the number of graph neural network layers on the performance of AGNN-ILM on UMLS. Best performances are shown in bold, and the underlined values indicate the second best.

Comparison	UMLS		
	MRR	**H@1**	**H@10**
n = 5	<u>0.984</u>	<u>0.976</u>	<u>0.996</u>
n = 6	**0.985**	**0.978**	**0.998**
n = 7	0.977	0.966	<u>0.996</u>
n = 8	0.975	0.961	0.995
n = 9	0.979	0.969	0.995

The training curves of AGNN-ILM employing graph neural networks with varying numbers of layers are illustrated in Figure 10. The training curves of AGNN-ILM indicate that as the number of layers in the graph neural network increases, the convergence of the model gradually decreases. The performance of AGNN-ILM reaches equilibrium after 40 epochs when $n = 5$. However, the model performance reaches equilibrium after 200 epochs when $n = 9$. The larger the number of layers in graph neural networks, the slower the convergence speed of the model. Increasing the number of network layers does not always enhance performance. On the contrary, it may result in performance degradation. Therefore, selecting an appropriate number of network layers is crucial to ensuring the performance of the model.

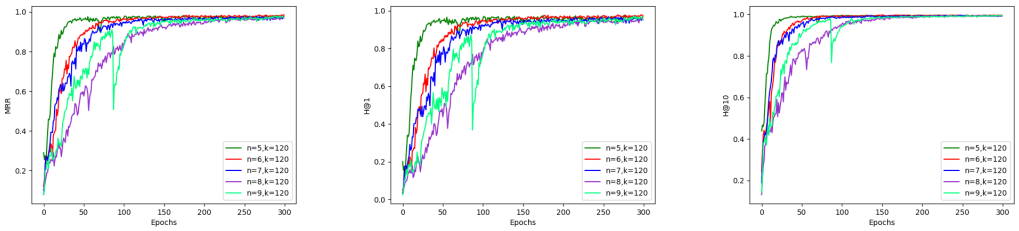


Figure 10. Experimental results on the effect of the number of graph neural network layers on the performance of AGNN-ILM on UMLS dataset.

5. Conclusions

An efficient model for learning propagation paths adaptively in knowledge graph reasoning tasks, AGNN-ILM, is proposed. Existing graph neural network-based approaches overlook the semantic correlation between the local neighborhoods. AGNN-ILM utilizes a semantic-aware distribution to choose local neighborhoods intelligently. In addition, the incremental learning mechanism improves the sampling efficiency of AGNN-ILM by preserving layer-wise connection, which can retain the relevant entities in the propagation path. At the same time, the parameter initialization technique is employed to enhance the stability of AGNN-ILM. The experimental results demonstrate that AGNN-ILM achieves excellent performance in knowledge graph reasoning tasks.

In future work, we will focus our research on a combined model of transformer and a graph neural network to enhance the information propagation capability of the graph neural network. Due to the powerful comprehension ability of large language models, we will explore integrating the symbolic reasoning capability of large models into knowledge graph reasoning tasks in our future research work.

Author Contributions: Conceptualization, J.Z. and H.Z.; methodology, J.Z.; software, J.Z., S.W. and J.H.; validation, J.Z., S.W. and J.H.; investigation, J.Z., S.W. and J.H.; data curation, J.Z., S.W. and J.H.; writing—original draft preparation, J.Z., S.W. and J.H.; writing—review and editing, J.Z., S.W. and H.Z.; visualization, J.Z., S.W., K.Z. and J.H.; supervision, J.Z., K.Z. and H.Z.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China (Grant No. U23A20316).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Authors Junhui Zhang, Shuning Wu and Jianwei Huo are employed by the company Yunhe (Henan) Information Technology Co., Ltd. Authors Junhui Zhang and Jianwei Huo are employed by the company Yellow River Engineering Consulting Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G.D.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge graphs. *ACM Comput. Surv.* **2021**, *54*, 1–37. [CrossRef]
2. Du, Z.; Zhou, C.; Yao, J.; Tu, T.; Cheng, L.; Yang, H.; Zhou, J.; Tang, J. CogKR: Cognitive Graph for Multi-Hop Knowledge Reasoning. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 1283–1295. [CrossRef]
3. Opdahl, A.L.; Al-Moslmi, T.; Dang-Nguyen, D.T.; Gallofré Ocaña, M.; Tessem, B.; Veres, C. Semantic Knowledge Graphs for the News: A Review. *ACM Comput. Surv.* **2022**, *55*, 1–38. [CrossRef]
4. Nie, W.; Bao, Y.; Zhao, Y.; Liu, A. Long Dialogue Emotion Detection Based on Commonsense Knowledge Graph Guidance. *IEEE Trans. Multimed.* **2024**, *26*, 514–528. [CrossRef]
5. Zhao, Y.; Wang, X.; Chen, J.; Wang, Y.; Tang, W.; He, X.; Xie, H. Time-aware Path Reasoning on Knowledge Graph for Recommendation. *ACM Trans. Inf. Syst.* **2022**, *41*, 1–26. [CrossRef]
6. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.

7. Li, Z.; Liu, X.; Wang, X.; Liu, P.; Shen, Y. TransO: A knowledge-driven representation learning method with ontology information constraints. *World Wide Web* **2022**, *26*, 297–319. [CrossRef]
8. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019; Volume 32.
9. Zhang, S.; Tay, Y.; Yao, L.; Liu, Q. Quaternion knowledge graph embeddings. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
10. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, E.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 Jun 2016; Volume 48, pp. 2071–2080.
11. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 1–8.
12. Shen, Y.; Ding, N.; Zheng, H.T.; Li, Y.; Yang, M. Modeling Relation Paths for Knowledge Graph Completion. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 3607–3617. [CrossRef]
13. Lao, N.; Cohen, W.W. Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.* **2010**, *81*, 53–67. [CrossRef]
14. Xiong, W.; Hoang, T.; Wang, W.Y. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 564–573.
15. Wang, H.; Ren, H.; Leskovec, J. Relational Message Passing for Knowledge Graph Completion. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 14–18 August 2021; pp. 1697–1707.
16. Zhu, Z.; Yuan, X.; Galkin, M.; Xhonneux, L.P.; Zhang, M.; Gazeau, M.; Tang, J. A*Net: A Scalable Path-based Reasoning Approach for Knowledge Graphs. In Proceedings of the Advances in Neural Information Processing Systems, Changsha, China, 20–23 November 2023; Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2023; Volume 36, pp. 59323–59336.
17. Ding, Y.; Yao, Q.; Zhao, H.; Zhang, T. DiffMG: Differentiable Meta Graph Search for Heterogeneous Graph Neural Networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 14–18 August 2021; pp. 279–288.
18. Xiao, W.; Zhao, H.; Zheng, V.W.; Song, Y. Neural PathSim for Inductive Similarity Search in Heterogeneous Information Networks. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, New York, NY, USA, 1–5 November 2021; pp. 2201–2210.
19. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef] [PubMed]
20. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In *The Semantic Web, Proceedings of the 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018*; Proceedings 15; Springer: Cham, Switzerland, 2018; pp. 593–607.
21. Yu, D.; Yang, Y.; Zhang, R.; Wu, Y. Knowledge embedding based graph convolutional network. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 1619–1628.
22. Zhang, Y.; Yao, Q. Knowledge graph reasoning with relational digraph. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 912–924.
23. Yoon, M.; Gervet, T.; Shi, B.; Niu, S.; He, Q.; Yang, J. Performance-adaptive sampling strategy towards fast and accurate graph neural networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 2046–2056.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
25. Yang, F.; Yang, Z.; Cohen, W.W. Differentiable learning of logical rules for knowledge base reasoning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2316–2325.
26. Sadeghian, A.; Armandpour, M.; Ding, P.; Wang, D.Z. Drum: End-to-end differentiable rule mining on knowledge graphs. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 15347–15357.
27. Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; McCallum, A. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
28. Qu, M.; Chen, J.; Xhonneux, L.P.; Bengio, Y.; Tang, J. RNNLogic: Learning Logic Rules for Reasoning on Knowledge Graphs. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
29. Cheng, K.; Liu, J.; Wang, W.; Sun, Y. RLogic: Recursive Logical Rule Learning from Knowledge Graphs. In Proceedings of the 2022 ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'22), Washington, DC, USA, 14–18 August 2022.
30. Qu, M.; Tang, J. Probabilistic logic neural networks for reasoning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 7712–7722.
31. Vashishth, S.; Sanyal, S.; Nitin, V.; Talukdar, P. Composition-based Multi-Relational Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.

32. Xu, X.; Feng, W.; Jiang, Y.; Xie, X.; Sun, Z.; Deng, Z.H. Dynamically Pruned Message Passing Networks for Large-scale Knowledge Graph Reasoning. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
33. Zhu, Z.; Zhang, Z.; Xhonneux, L.P.; Tang, J. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 29476–29490.
34. Zhang, Y.; Zhou, Z.; Yao, Q.; Chu, X.; Han, B. Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 3446–3457.
35. Lee, J.; Chung, C.; Whang, J.J. InGram: Inductive knowledge graph embedding via relation graphs. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 18796–18809.
36. Huang, X.; Romero, M.; Ceylan, I.; Barceló, P. A Theory of Link Prediction via Relational Weisfeiler-Leman on Knowledge Graphs. In Proceedings of the Advances in Neural Information Processing Systems, Changsha, China, 20–23 November 2023; Curran Associates, Inc.: Red Hook, NY, USA, 2023; Volume 36, pp. 19714–19748.
37. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; Gordon, G., Dunson, D., Dudík, M., Eds.; Volume 15, pp. 315–323.
38. Jang, E.; Gu, S.; Poole, B. Categorical Reparametrization with Gumble-Softmax. In Proceedings of the International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.
39. Kok, S.; Domingos, P. Statistical predicate invention. In Proceedings of the 24th International Conference on Machine Learning, New York, NY, USA, 20–24 June 2007; pp. 433–440.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.