HiHo: A Hierarchical and Homogenous Subgraph Learning Model for Knowledge Graph Relation Prediction

Jiangtao Ma [a,b], Yuke Ma [c], Fan Zhang [d], Yanjun Wang [b], Xiangyang Luo [e], Chenliang Li [f] and Yaqiong Qiao [g,*]

[a] *School of Computer and Information Engineering, Tianjin Normal University, 300387, Tianjin, China*
E-mail: majiangt@139.com
[b] *School of Computer Science and Technology, Zhengzhou University of Light Industry, 450000, Zhengzhou, China*
E-mails: majiangt@139.com, wyj@zzuli.edu.cn
[c] *Institute of Information Technology, Information Engineering University, 450002, Zhengzhou, China*
E-mail: mayuke0418@qq.com
[d] *China National Digital Switching System Engineering and Technology R&D Center, 450001, Zhengzhou, China*
E-mail: 17034203@qq.com
[e] *State Key Laboratory of Mathematical Engineering and Advanced Computing, 450001, Zhengzhou, China*
E-mail: xiangyangluo@126.com
[f] *School of Cyber Science and Engineering, Wuhan University, 430079, Wuhan, China*
E-mail: lichenliang.whu@gmail.com
[g] *College of Cyber Science, Nankai University, 300350, Tianjin, China*
E-mail: 9920240077@nankai.edu.cn

Abstract. Relation prediction in Knowledge Graphs (KGs) aims to anticipate the connections between entities. While both transductive and inductive models are incorporated for context comprehension, we need to focus on two primary issues. First, these models only collate relations at each layer of the subgraph, overlooking the potential sequential relationship between different layers. Second, these methods overlook the homogeneity of subgraphs, thus impeding their ability to effectively learn the importance of relationships within the subgraphs. To address this challenge, we propose a hierarchical and homogenous subgraph learning model for knowledge graph relation prediction (HiHo). Specifically, we adopt a subgraph-to-sequence mechanism (S2S) to learn the potential semantic associations between layers in the subgraph of a single entity, and thus model the hierarchy of the subgraph. Then, we implement a common preference inference mechanism (CPI) that assigns higher weights to co-occurrence relations while learning the importance of each relation in the subgraphs of two entities, and thus model the homogeneity of the subgraph. In our study, we sequentially employ induction on each layer of subgraphs pertaining to the two entities for relation prediction. To assess the efficacy of our method, we perform experiments on five publicly available datasets. The results of our experiments demonstrate that our method surpasses the current state-of-the-art baselines in both transductive and inductive settings.

Keywords: knowledge graph, relation prediction, inductive method

*Corresponding author. E-mail: 9920240077@nankai.edu.cn.

1. Introduction

Knowledge graphs (KGs) are structured collections of facts represented as triples (head entity, relation, tail entity), playing a fundamental role in numerous natural language processing tasks, such as multi-hop query [1], question answering [2], recommendation systems [3], commonsense reasoning [4]. Despite the success, most of the KGs often suffer from incompleteness, especially the missing relations between entities. Therefore, an increasing number of researchers are commited to predict the missing relations between entities [5], and many different methods have been proposed. These methods can be roughly categorized into transductive methods and inductive methods.

Transductive methods refers to the traditional KG embedding methods, which can be divided into translation-based methods (e.g., TransE [6], TransR [7]), semantic matching methods(e.g., ComplEx [8], DistMult [9]), path-based methods(e.g., CURL[10], CPL [11]), graph-based methods(e.g., TransEQ [12], NoGE [13]). Despite their different forms, transductive methods mostly focus on learning matching score for the relations of known inherent entities in KG. Many transductive methods have proven to be effective in predicting relations between entities, but they need expensive re-training to infer the relations between unknown new entities, which is unpractical for real-world KGs that are constantly adding new entities.

Different from transductive methods, induction-based methods aim at learning the subgraph structure of entities to infer the relations between entities in KGs, where entities may be unseen in the training process. For instance, Grail [14] is a framework based on Graph Neural Networks (GNN) with strong inductive bias, which can learn relationship semantics unrelated to entities. Grail does not learn embeddings of specific entities, but predicts relationships from subgraph structures around candidate relationships. SNRI [15] incorporates complete relational information into enclosing sparse subgraph in a global way for relation prediction. TACT [16] predicts the missing link by learning the topology-aware semantic correlations between relations in the entity-independent manner. However, these methods only collect relations at each layer of the subgraph individually, without considering the back-and-forth associations of relations in different layers, so they lose the global nature of the relations. Furthermore, they approximate the commonalities between subgraphs by simply extracting their intersection, and they prune nodes that are isolated or located at a distance greater than a specified threshold (denoted as "k") from either of the target entities. But sometimes the intersection occupies only a small portion of the two subgraphs, which leads to many relation in the subgraphs being ignored.

To solve the above problem, we first reviewed the subgraphs of entities. Through our observation, we found that the subgraph of a entity consists of the neighboring entities and relations surrounding the entity. These neighboring entities and neighboring relations exist in the form of layers, and any level of the subgraph does not exist in isolation. For the-th layer of the subgraph, both its forward and backward layers can provide potential information for it. The assumption we are based on is that the process of human understanding things usually includes understanding the history of things and predicting their future. So for machines, they should cognize the history and future information of i-th layer of the subgraph to model it. Therefore, we assume that combining the back-and-forth association of each layer of the subgraph should be helpful for relation prediction. Furthermore, we found that the intersection of the subgraphs of the head and tail entity, i.e., the co-occurrence relations in the two subgraphs, can provide valuable information for predicting the relation that are connected to both the head and tail entity. These co-occurrence relations are more likely to be true relations than other relations, so they should be given higher weight. We operate under the assumption that the higher the frequency of a relation (denoted as r_i) appearing in the intersection of the subgraphs of the head and tail entities, the more likely it is that r_i is simultaneously associated with both entities. Consequently, r_i is more likely to represent the true relation between the head and tail entities and should thus be given greater preference. However, it's essential to recognize the significance of other relations as well.

To implement the above process, we propose a subgraph-to-sequence mechanism (Sg2S) that converts the subgraphs of a entity into a sequences composed of layers, specifically we choose Bi-GRU to encode the history and future information for the layers in the sequence to model the hierarchical nature of the subgraphs. Furthermore, we propose a common preference inference mechanism (CPI) that assigns higher weights to co-occurrence relations while learning the importance of each relation to both subgraphs. Also, to prevent over-preference for certain relations, a smoothing factor is added to CPI for balancing the preferred and unpreferred relations. In this way, each neighbor relations is assigned a weight to indicate the importance of the relation in the two subgraphs. Finally, we

alternately induct each layer of subgraph of the head entity and tail entity to predict the relation between them. In summary, our main contributions are as follows.

• We introduce a Subgraph-to-Sequence mechanism (S2S) to capture the hierarchical structure within the subgraph of an individual entity. This mechanism facilitates the understanding of the entity's internal relationships and context.

• We propose a Common Preference Inference mechanism (CPI) to address the homogeneity within the subgraphs of the head and tail entities. CPI aims to model and leverage the shared characteristics present in both subgraphs, enhancing the predictive accuracy of relations between entities.

• We propose HiHo to predict the relations between entities by modeling the hierarchy and homogeneity of the subgraphs of entities. In this way, HiHo avoids learning to embed any specific entity, so it can predict the relations between emerging or unseen entities in the reasoning phase.

2. Related Work

Recently, researchers focused on predicting relationships within Knowledge Graphs (KGs) because it's essential for many tasks. As a result, more scholars are now studying this area. Existing KG relation prediction methods fall into two main categories: transductive and inductive methods.

2.1. Transductive method

Transductive methods primarily concentrate on learning representations of entities and relations, employing a score function to determine the likelihood of a relation being associated with an entity [17]. These methods encompass various categories, and the most popular ones are translation-based methods, semantic matching methods, path-based methods, graph neural network (GNN)-based methods and graph-based methods.

Translation-based methods, such as TransE [6], TransR [7], TransH [18], ExpressivE [19], ConvRot [20] usually project entities and relations into the low dimensional feature spaces, and assume that the distance between the embeddings of the tail entity and head entity is approximately equal to the relation embedding. Despite their simplicity and efficiency, they ignore the features in the multi-layered neighborhoods of entities, which limits their ability to infer relations between entities.

Semantic matching methods, such as ComplEx [8], DistMult [9], QuatE [21], DRUM [22] employs a scoring function grounded in similarity rules, building upon the framework of translation-based methods. Through this approach, they gauge the likelihood of connections between relations and entities by aligning the semantic associations of entities and relations within the embedded vector space. Nonetheless, their effectiveness heavily relies on the configuration of rules, and certain missing relations may remain uninferrable through any rule [23].

Path-based methods, such as P-INT [24], AstarNet [25], HiAM [26], CURL[10], CPL [11] usually define a path as the sequence of relations and select relevant features from a path, then model the interactions between the paths to infer new relations between entities. However, in many KGs, the length and number of paths are not balanced, and some paths that are too much, too little, too long, or too short can affect the performance of relation reasoning.

Graph-based methods usually design a graph neural network to learn the subgraph structure of entities and obtain the representation of entities. On the basis of GAT [27], EIGAT [28], GGAE [29] assign distinct attention weights to the neighbor entities or the neighbor relations of the central entity to obtain entity representation. And on the basis of GCN[30], CompGCN[31], KE-GCN [32], DA-GCN [33] model the node or relationship features in the full graph by convolution operations to obtain the entity representation. Although their implementation details are different, they all aggregate the neighboring features of nodes into the node representation [34], which is the reference for relation prediction.

However, these methods can only model known entities and predict relations between known entities during training. Therefore, they require expensive re-training to model the constantly added unknown entities in KGs, and to infer the relations between unknown entities and the relations between unknown and known entities, which makes them difficult to be applied to the ever-changing KGs.

2.2. Inductive method

Inductive methods usually infer relations between entities from the subgraphs around entities, avoiding learning to embed any specific entity. They can be naturally generalized to unseen entities or emerging entities in KG, as they learn to reason over subgraph structures independent of any specific entity representation. Recently, inductive methods have been receiving more and more attention. They are classified into rule-based and subgraph-based methods.

Rule-based methods involve the explicit learning of logical rules for reasoning, which are independent of entities and thus considered inductive. Yang et al. [35] introduced NeuralLP, the first end-to-end differentiable model inspired by TensorLog and neural networks, to learn variable rule lengths. This model merges first-order rule reasoning with sparse matrix multiplication and introduces a neural controller system featuring attention mechanisms and memory, enabling simultaneous learning of the parameters and structure of first-order logical rules. However, this model is constrained by the maximum length of rules and may inadvertently extract incorrect rules with high confidence. Sadeghian et al. [22] improved Neural LP with the introduction of Drum, where the learning confidence scores for each rule are related to the low-rank tensor approximation. They use BiRNN to share useful information across different relation learning tasks. However, they often overlook the structure surrounding the target triple, leading to limited expressive ability.

The method based on subgraphs can utilize structural information for inductive reasoning. GraIL [14] is the first method to learn entity-independent relational semantics to predict the relation between two entities by reasoning about the local subgraph structure around the two entities. TACT [16] transforms a KG to a relational correlation graph, and proposes a relational correlation network to learn topology-aware correlations between relations in entities' neighborhoods by a entity-independent manner, to predict the missing relation. CFAG [36] utilizes a coarse-grained module to generalize the unseen entities with multiple relational semantic, then uses a fine-grained module to generate more accurate entity representations with certain query relations. MorsE [37] utilizes an entity initializer to generate each entity's initial embedding through relation-domain embedding settings and relation-range embedding settings, then learns the neighbor structure of the entity to enhance entity embedding by a GNN modulator. Next, MorsE resorts to meta-learning to output an entity's final embeddings. These methods model the rules between relations to generate relation preferences for entities. PathCon [38] proposes a multi-layer relational message passing mechanism, which iteratively aggregates the relational context features of entities and the relational path features between entities to predict missing relations. SNRI [15] extracts complete neighboring relations for each entity and constructs neighboring relational paths by the mutual information maximization mechanism. By combining the above two approaches, SNRI can effectively integrate the comprehensive relational information into the subgraph of entities to improve the performance of relation prediction. StATIK [39] aggregates structure information from the neighborhoods of entities through a message passing neural network, then completes the missing relations by combining underlying textual descriptions of entities and relations through a pre-trained language model. RMPI [40] utilizes novel techniques such as target relation-guided graph pruning strategies, target relation-aware neighborhood attention, handling empty subgraphs, and ontology-based relationship semantic injection to perform relation prediction through relation message passing. These methods are not simply graph-based iterative methods, they all choose to combine other methods to obtain a better relation prediction effect.

These methods infer the relation between two entities by generalizing the subgraphs around those two entities. However, these methods neglect the heterogeneity and hierarchy natures of subgraphs, which limits their performance in relation prediction.

3. Problem Definition

Definition 1 (Knowledge Graph): $\mathcal{G}(\mathcal{E},\mathcal{R}) = \{(h,r,t) \mid h,t \in \mathcal{E}, r \in \mathcal{R}\}$. $\mathcal{E}$ represents a set of entities, $\mathcal{R}$ represents a set of relations, and $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$ represents a set of fact triplets. A fact triplet is represented as (h,r,t), where h denotes the head entity, t denotes the tail entity, and r denotes the relationship between the head and tail entities. A fact triplet such as (Heartbleed Vulnerability, exists in, OpenSSL Program), where "Heartbleed Vulnerability",

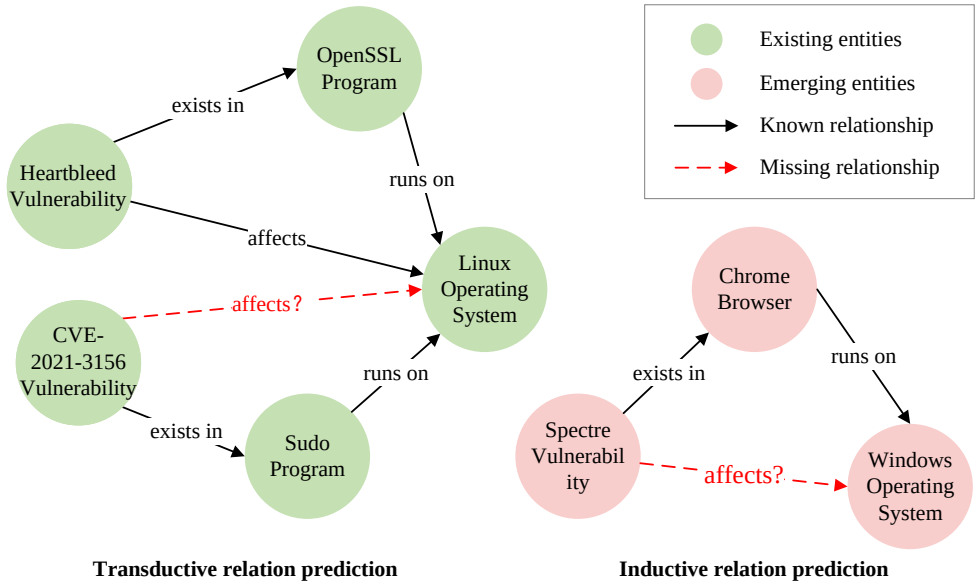

Fig. 1. **Illustration of the transductive and inductive settings for relation prediction in knowledge graphs.** The transductive relation prediction method can only identify known entities, while inductive relation prediction is learned from the training knowledge graph (KG) to infer in another KG without shared entities. Green entities represent known entities during training, red entities represent newly emerged entities, and the red dashed lines indicate the relationships to be predicted.

"exists in", and "OpenSSL Program" are respectively the head entity, relationship, and tail entity, represents the fact that the heartbleed vulnerability exists in the openSSL program.

Definition 2 (Inductive Knowledge Graph): An inductive knowledge graph, $\mathcal{G}'(\mathcal{E}', \mathcal{R})$, consists of a set of invisible entities, $\mathcal{E}'$, and a set of visible relations, $\mathcal{R}$, which are shared with the original knowledge graph. $\mathcal{G}'(\mathcal{E}', \mathcal{R}) = \{(h, r, t) \mid h, t \in \mathcal{E}', r \in \mathcal{R}\}$. $\mathcal{E}' \cap \mathcal{E} = \varnothing$, where $\mathcal{E}'$ represents a set of invisible entities.

Definition 3 (Knowledge Graph Relation Prediction): Knowledge graph relation prediction aims to train models to predict missing relationships given a triplet $(h, ?, t)$.

Definition 4 (Transductive Relation Prediction): Transductive relation prediction is defined as predicting missing triplets: $\mathcal{F}^1 = \{(h, r, t) \mid (h, t \in \mathcal{E}, r \in \mathcal{R})\}$.

Definition 5 (Inductive Relation Prediction): In the inductive setting, relation prediction aims to quantify the score of each relation r in the graph $\mathcal{G}$ and predict the relationship between two invisible entities h and t in the test data. Inductive relation prediction is defined as predicting the missing triplet $\mathcal{F}^2 = \{(h, r, t) \mid h, t \in \mathcal{E} \cup \mathcal{E}'', r \in \mathcal{R} \cup \mathcal{R}''\}$, where $\mathcal{E}''$ represents a set of invisible entities, and $\mathcal{R}''$ denotes a set of hidden relations. As shown in Fig. 1, in the inductive relation prediction of KG, the triplet (invisible head entity: Chrome Browser, visible relationship: runs on, invisible tail entity: Windows Operating System) in the emerging KG shares the visible relationship "runs on" with the triplet (visible head entity: OpenSSL Program, visible relationship: runs on, visible tail entity: Linux Operating System) in the original KG. However, all triplets in the original KG are visible, while entities in the emerging KG are all invisible.

4. Proposed Method

For a given entity pair (h, t), where h is the head entity and t is the tail entity. When inferring the relation r between h and t, the subgraph around h and subgraph around t are very important. They can reveal the properties of h and t [38], and provide logical evidence to infer the relationship type between h and t [14]. However, there are a large number of discrete atomic symbols in these subgraphs, so it is almost impossible to directly model them [41].

To address this issue, we propose a new inductive relation prediction method. Specifically, we design Subgraph-to-Sequence (S2S) to infer the hierarchical information of a single subgraph, The GRU processes the state sequences

within the S2S mechanism, allowing the model to capture the contextual relationships between the states and thus obtain the global characteristics of the subgraph. Then, we propose a common preference inference (CPI) mechanism to learn the homogenous information between two subgraphs, and reduce the differences between subgraphs by strengthening their common parts, so as to obtain the local characteristics of subgraphs. The CPI assigns different weights to relations. This weighted information from the CPI is then used as part of the input features when the GRU in the S2S processes the state sequences. The GRU, influenced by the CPI's weighted relations, further refines its capture of the hierarchical information within the subgraph layers. Moreover, this paper proposes an alternate induction method, which collects the relations in the subgraph of a head entity and the subgraph of a tail entity alternately to infer the relation between entities.

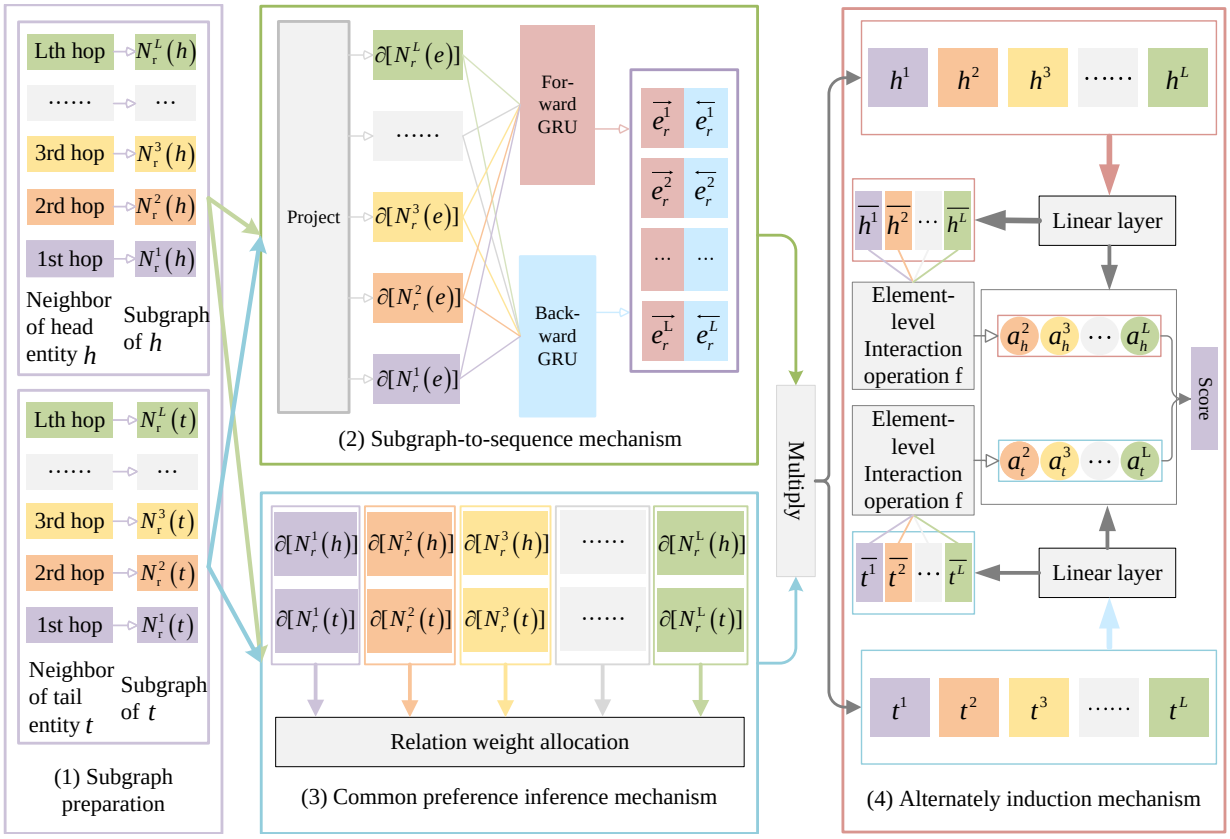

Fig. 2. The framework diagram of the proposed method, HiHo, comprises four components: (1) Subgraph Preparation selectively samples up to K neighbor relations from each entity's neighborhood into the subgraph; (2) Subgraph-to-Sequence Mechanism (S2S) used to capture the semantic associations between layers within the subgraph of a single entity and achieve hierarchy; (3) Common Preference Inference Mechanism (CPI) which assigns higher weights to co-occurrence relations and learns the importance of each relation in the subgraphs of two entities; (4) Alternating Induction Mechanism that learns the subgraphs of two entities to predict relations between them.

4.1. Subgraph Preparation

Unlike traditional methods that simultaneously collect entities and relations from the neighborhoods of entities into subgraphs, we only collects relations from each entity's neighborhood into the subgraph. However, the relations in the neighborhood of some entities are too dense to be fully used. For example, we observe that there are thousands of relations in the 3rd hop neighborhoods of some entities in NELL995 dataset, while the 1st hop neighborhoods of some entities in some real-world KGs may encompass many thousands of relations, so it is impractical to store and model them all.

The solution to this problem is that, for entity e, we first initialize its subgraph to be empty. Then, starting from the 1st hop neighborhood of entity e, we sample up to K neighbor relations as the 1st layer of the subgraph, and add them to the subgraph. Next, for the entities connected with these neighbor relations, we sample up to K neighbor relations of each entity as the 2nd layer of the subgraph and add them to the subgraph, and so on, lasting for L times. Eventually, each entity's subgraph contains no more than K^L neighbor relations. Based on this, for entity e, we obtain its subgraph $N_r(e) = \left\{N_r^1(e), N_r^2(e), \ldots, N_r^L(e)\right\}$.

4.2. Subgraph-to-Sequence Mechanism

The subgraph of the entity contains a vast number of relations. These relations exist in the form of layers, and any layer of the subgraph does not exist in isolation. Taking Figure 2 as an example, for the entity "Heartbleed Vulnerability", each layer of its subgraph does not exist independently and has a semantic association with other layers. For instance, the 1st layer of "Heartbleed Vulnerability" contains the relation "exists in", and we can infer that the type of the unknown tail entity 1 connected to "Heartbleed Vulnerability" may be "affects". And the 2nd layer contains the relation "runs on", and we can infer that the type of the unknown tail entity 2 may be a larger range of location. By learning the semantic association between the 1st layer and the 2nd layer in the subgraph, it can be inferred that the relation between "Heartbleed Vulnerability" and the unknown tail entity 2 may be "affects". Thus, modeling the hierarchical nature of subgraphs, i.e., learning the potential semantic associations between layers in individual subgraphs, can generate more accurate relation tendencies for entities.

Similar to Graph2Seq [42], we regard the subgraph $N_r(e)$ as a sequence consisting of $N_r^i(e)$ where $i \in [1, L]$. Here, "e" represents the entity for which we are constructing and analyzing the subgraph. In the multi-hop subgraph sequence related to the entity , the forward layers $N_r^{i-k}(e)$ (the layers that come before a certain point in the sequence) provide historical information, while the backward layers $N_r^{i+k}(e)$ (the layers that come after that point) supply future information.

Next, we opted to use the Bidirectional Gated Recurrent Unit (Bi-GRU), which is a time series prediction method based on the Gated Recurrent Unit (GRU). It combines the bidirectional model and the gating mechanism. The overall structure and the cell structure are consistent with those of GRU, thus enabling it to effectively capture the temporal relationships in time series data. The overall structure of Bi-GRU consists of two GRU networks in opposite directions. One network processes the time series data from the front to the back, and the other network processes it from the back to the front. This bidirectional structure can simultaneously capture both past and future information, thereby more comprehensively modeling the temporal relationships in the time series data. We use Bi-GRU to encode the features from the previous and later parts of the sequence for each state. This enables us to capture the contextual relationships between states, which is equivalent to the hierarchical information present between each layer of the subgraph. The output of the Bi-GRU can be described by Eq. (1) and Eq.(2):

$$\overrightarrow{e_r^i} = \text{GRU}\left(\partial\left[N_r^1(e)\right], \partial\left[N_r^2(e)\right], ..., \partial\left[N_r^i(e)\right]\right) \tag{1}$$

$$\overleftarrow{e_r^i} = \text{GRU}\left(\partial\left[N_r^i(e)\right], \partial\left[N_r^{i+1}(e)\right], ..., \partial\left[N_r^L(e)\right]\right) \tag{2}$$

$$e_r^i = \left[\overrightarrow{e_r^i}, \overleftarrow{e_r^i}\right] \tag{3}$$

where sequences $\overrightarrow{e_r^i}$ and $\overleftarrow{e_r^i}$ are the forward and backward hidden layer state sequences which are the outputs of the Bi-GRU. Thus, the layer state sequence stitching e_r^i is generated, $i \in [1, L]$ is the layer index, as shown in Eq.(3). $\partial(\cdot)$ projects $N_r^i(e)$ as the vector by representing each relation in the subgraph as a one-hot identity vector of the relation type that its belongs to, as follows:

$$\partial\left[N_r^i(e)\right] = \{x_1, x_2, ..., x_m\}, x_j = one - hot(r_j) \tag{4}$$

where x_j is the one-hot identity vector of r_j, $j \in [1, m]$, and $m \leqslant K$ is the number of relations in $N_r^i(e)$.

By the above way, Subgraph-to-Sequence (S2S) transforms the subgraph $N_r(e) = \left\{N_r^1(e), N_r^2(e), \ldots, N_r^L(e)\right\}$ into the state sequence $S_r(e) = \left\{e_r^1, e_r^2, e_r^3, \ldots, e_r^L\right\}$.

4.3. Common Preference Inference Mechanism

To better capture the homogeneity between the i-th layer subgraphs $N_r^i(h)$ and $N_r^i(t)$ of entities h and t, based on the assumption that the more frequent the r_i appears in the intersection of $N_r^i(h)$ and $N_r^i(t)$, we introduce a common preference inference mechanism to infer the importance of each relation in $N_r^i(h)$ and $N_r^i(t)$, and assign higher weights to the co-occurrence relations in $N_r^i(h)$ and $N_r^i(t)$. We further introduce a smoothing factor λ to balance the preferred and unpreferred relations and prevent over-preference for certain relations. Take Figure 1 as an example, the 1st layer in the subgraphs of both entity "OpenSSL Program" and entity "Sudo Program" contain the same relation "runs on", and both entities are connected to the tail entity with "Linux Operating System", then relation "runs on" may be the real relation between these two entities, so it should be given a higher weight.

Specifically, for $N_r^i(h)$ and $N_r^i(t)$ we learn the homogeneity between them and obtain the latent subgraph representation of them, as shown in Eq.(5) and Eq.(6).

$$h^i = h_r^i \cdot g\left(N_r^i(h), N_r^i(t)\right) \tag{5}$$

$$t^i = t_r^i \cdot g\left(N_r^i(h), N_r^i(t)\right) \tag{6}$$

where h_r^i and t_r^i denote the layer state sequence of $N_r^i(h)$ and $N_r^i(t)$, as shown in Eq.(3). And $g(N_r^i(h), N_r^i(t))$ is the relation weight matrix, containing the weights of each relation, as shown in Eq.(7).

$$g\left(N_r^i(h), N_r^i(t)\right) = \lambda \frac{N_r^i(h) \cap N_r^i(t)}{N_r^i(h) \cup N_r^i(t)} + (1-\lambda)\frac{N_r^i(h) \cup N_r^i(t) - N_r^i(h) \cap N_r^i(t)}{N_r^i(h) \cup N_r^i(t)} \tag{7}$$

$$g\left(N_r^i(h), N_r^i(t)\right) \leftrightarrow \{w_1, w_2, w_3, ..., w_n\} \tag{8}$$

where λ is a smoothing factor to balance the preferred co-occurrence relations with the other relations. In this formula, a plus sign connects two terms, the first term emphasizes the common part between the two subgraphs. The second term represents the difference between the two subgraphs. By subtracting the intersection from the union and normalizing it, we get a measure of the unique parts of each subgraph. When λ is larger, the model gives more favor to co-occurrence relations; the smaller λ is, the model gives more favor to other relations. Intuitively, for different subgraphs, there exist different optimal λ. After extensive experimentation and evaluation on the benchmark datasets, we found that a lambda value of 0.5 yielded the best overall performance in most cases.

Finally, the common preference inference mechanism functions by assigning different weights w_i to each relation r_i. These weights are calculated based on the significance of r_i within the subgraphs $N_r^i(h)$ and $N_r^i(t)$ associated with the head entity h and the tail entity t. The more important a relation r_i is in these subgraphs, the higher the weight w_i it will be assigned. This weighting allows us to focus on the relations that are more relevant in characterizing the entities h and t. By emphasizing these important relations, we are able to capture more similar features between h and t. For example, if a particular relation r_i frequently appears in both $N_r^i(h)$ and $N_r^i(t)$, it will likely have a higher weight, indicating its importance in determining the similarity between the two entities. This, in turn, helps us to reason about the possible relation between h and t. As shown in Eq.(8), this mechanism is formalized in a way that enables the model to effectively utilize these weighted relations for better inference.

4.4. Alternating Induction Mechanism

After learning the hierarchical features and homogeneous features of each layer of subgraphs of the head entity and tail entity, we alternately induct the i-th layer subgraph of the head and tail entities to obtain the inductive subgraph representations of the head and tail entities, as shown in Eq. (9) and Eq. (10).

$$a_h^i = \sum_{i\in[2,L]} f\left(\overline{h^i}, \overline{h^{i-1}}\right) \tag{9}$$

$$a_t^i = \sum_{i\in[2,L]} f\left(\overline{t^i}, \overline{t^{i-1}}\right) \tag{10}$$

$$\overline{h^i} = \sigma\left[\left(a_h^{i-1} + h^i + a_t^{i-1}\right) W_h^i + b\right], i \in [2, L] \tag{11}$$

$$\overline{t^i} = \sigma\left[\left(a_h^{i-1} + t^i + a_t^{i-1}\right) W_t^i + b\right], i \in [2, L] \tag{12}$$

where a_h^i and a_t^i denote the latent semantic features of h and t at the i-th alternate induction. h^i and t^i denote the latent subgraph representation, as shown in Eq. (5) and Eq. (6). $\overline{h^i}$ and $\overline{t^i}$ denote the inductive representation of the i-th subgraph of h and t at the i-th iteration. And in Eq. (9) and (10), we multiply the current inductive representation by the previous one layer inductive representation to learn the direct historical information [26]. $\sigma(\cdot)$ is a sigmoid function, $f(\cdot)$ is the element-level interaction operation, as shown in Eq. (13).

$$f\left(e^i, e^{i-1}\right) = \begin{pmatrix} \left(e^i\right)^{(1)} \cdot \left(e^{i-1}\right)^{(1)} & \cdots & \left(e^i\right)^{(1)} \cdot \left(e^{i-1}\right)^{(d)} \\ \vdots & \ddots & \vdots \\ \left(e^i\right)^{(d)} \cdot \left(e^{i-1}\right)^{(1)} & \cdots & \left(e^i\right)^{(d)} \cdot \left(e^{i-1}\right)^{(d)} \end{pmatrix} \tag{13}$$

The final latent semantic feature a_h^L and a_t^L represent the inductive subgraph representations. They are further combined and input to the linear layer to obtain the score of (h, t), which is used to predict the relation type between h and t. The score is a crucial metric that is calculated as follows. It is obtained by taking the dot product of the latent semantic features a_h^L and a_t^L with the weight W, adding the bias b, and then applying the sigmoid function σ. This process is formalized in Eq. (14) where the score is calculated as Eq. (14).

$$Score(h,t) = \sigma\left[\left(a_h^L, a_t^L\right) \cdot W + b\right] \tag{14}$$

where W is the weight, b is bias. The score provides a quantitative measure that helps the model assess the likelihood or strength of a particular relation existing between the two entities, allowing for more accurate predictions and inferences about the relationships in the knowledge graph.

We train our method by minimizing the loss between the predicted relation type and the actual ground truth relation type, as shown in Eq. (15).

$$Loss = \sum_{(h_i,r_i,t_i)\in KG} C\left[Score\left(h_i, t_i\right), r_i\right] \tag{15}$$

where $C(\cdot)$ is the cross entropy loss function.

Algorithm 1: HiHo

Input: A set of entity pairs (h, t) in knowledge graph G.
Output: A set of relation R

1 Initialize the set of relation R
2 **for** *all entity pairs (h, t)* **do**
3 Function Subgraph Preparation:
4 **for** *all hop i in L* **do**
5 **for** *all hop k in K* **do**
6 $N_k^i(e) \leftarrow \{r_1, r_2, r_3, \ldots, r_K\}$ // Construct the k-hop neighbor set of entity e
7 **end**
8 $N_r(e) \leftarrow \left\{N_r^1(e), N_r^2(e), \ldots, N_r^L(e)\right\}$ // Aggregate neighbor sets for each layer
9 **end**
10 Function Subgraph-to-Sequence:
11 **for** *all hop i in L* **do**
12 **for** *entity e in* $N_r^i(e)$ **do**
13 // Concatenate the forward and backward hidden layer state sequences to get layer state sequence
14 $\overrightarrow{e_r^i} \leftarrow \mathrm{GRU}\left(\partial\left[N_r^1(e)\right], \partial\left[N_r^2(e)\right], \ldots\ldots, \partial\left[N_r^i(e)\right]\right)$
15 $\overleftarrow{e_r^i} \leftarrow \mathrm{GRU}\left(\partial\left[N_r^i(e)\right], \partial\left[N_r^{i+1}(e)\right], \ldots\ldots, \partial\left[N_r^L(e)\right]\right)$
16 $e_r^i \leftarrow \left[\overrightarrow{e_r^i}, \overleftarrow{e_r^i}\right]$
17 **end**
18 **end**
19 **end**
20 Function Common Preference Inference:
21 **for** *all hop i in L* **do**
22 $g\left(N_r^i(h), N_r^i(t)\right) = \lambda \frac{N_r^i(h) \cap N_r^i(t)}{N_r^i(h) \cup N_r^i(t)} + (1-\lambda) \frac{N_r^i(h) \cup N_r^i(t) - N_r^i(h) \cap N_r^i(t)}{N_r^i(h) \cup N_r^i(t)}$ // Calculate the relation weight matrix
23 $h^i = h_r^i \cdot g\left(N_r^i(h), N_r^i(t)\right)$ // Calculate the latent subgraph representation related to the head entity
24 $t^i = t_r^i \cdot g\left(N_r^i(h), N_r^i(t)\right)$ // Calculate the latent subgraph representation related to the tail entity
25 **end**
26 Function Alternating Induction:
27 **for** *all hop i in L* **do**
28 $a_h^i = \sum f\left(\bar{h}^i, \bar{h}^{i-1}\right)$ // Calculate the inductive representation of the head entity
29 $a_t^i = \sum f\left(\bar{t}^i, \bar{t}^{i-1}\right)$ // Calculate the inductive representation of the tail entity
30 $\bar{h}^i = \sigma\left[\left(a_h^{i-1} + h^i + a_t^{i-1}\right) W_h^i + b\right]$ // Update the state of the head entity
31 $\bar{t}^i = \sigma\left[\left(a_h^{i-1} + t^i + a_t^{i-1}\right) W_t^i + b\right]$ // Update the state of the tail entity
32 **end**
33 $r \leftarrow \mathrm{Score}(h,t) = \sigma\left[\left(a_h^L, a_t^L\right) \cdot W + b\right]$
34 Put r into R
35 Update $Loss = \sum_{(h_i, r_i, t_i) \in G} C\left[\mathrm{Score}\left(h_i, t_i\right), r_i\right]$
36 return R

4.5. Proposed Algorithm

The algorithm of HiHo is shown in Algorithm 1. First, we collect the relations in the neighborhood of h and t and add them to the subgraphs (lines 1-9). Then, we regard the subgraphs as the sequences, and use the Bi-GRU to encode previous feature and the later feature for each state in sequences (lines 10-17). Next, we learn the homogeneity between the subgraphs and obtain the latent subgraph representation of them (lines 19-24). Finally,

Table 1
Statistics of datasets

Datasets	Entities	Relations	Train Triples	Validations Triples	Test Triples	Total Triples
FB15K	14951	1345	483142	50000	59071	592213
FB15K-237	14541	237	272115	17535	20466	310116
NELL995	63917	198	137465	5000	5000	147465
WN18	40943	18	141442	5000	5000	151442
WN18RR	40943	11	86835	3034	3134	93003

we alternately induct the subgraph of h and t to obtain the score of (h, t) (lines 25-33), which is used to predict the relation type between h and t.

5. Experiments

In this section, we assess the performance of our proposed method across five benchmark datasets. The subsequent subsections outline the statistics of these benchmark datasets and summarize the parameter settings. Then, the experimental metrics chosen for this paper are presented. Next, we validate the effectiveness of our method through performance comparison and analysis in both the transductive relation prediction task and iterative relational prediction task. Subsequently, the effectiveness of different mechanisms on relation prediction is demonstrated. Finally, the influence of the parameters on the model is verified.

5.1. Datasets and experimental settings

The benchmark datasets selected in this paper are FB15K, FB15K-237, NELL995, WN18, and WN18RR, where FB15K, FB15K237 are extracted from Freebase [43], NELL995 are extracted from Nell [44], WN18, WN18RR are extracted from WordNet [45]. And in addition, FB15K-237, WN18RR are obtained by removing the inverse relations in FB15K and WN18 respectively. These datasets comprise numerous entities and relations, partitioned into training, validation, and test sets. Table 1 presents the statistics for these datasets.

We employ Adam [46] to train our method, the initial learning rate δ is selected within 0.001, 0.005, 0.01. At the same time, the embedding dimension of entities and relations in the triples is set to d, the value of which is selected from 16, 32, 64, 128, 256. Subsequently, the maximum number of layers in the neighborhood is set as $L_{hop} \in [1, 5]$, the maximum number of relations in each layer is set as $N_r \in [1, 10]$. Through grid search method, the optimal parameters of HiHo are finally selected as follows: δ =0.005, d=64, L_{hop} =2, N_r =8 on FB15K; δ =0.005, d=64, L_{hop} =2, N_r =8 on FB15K-237; δ =0.005, d=64, L_{hop} =2, N_r =8 on NELL995; δ =0.005, d=64, L_{hop} =3, N_r =8 on WN18; δ =0.005, d=64, L_{hop} =4, N_r =8 on WN18RR. We train at most 1000 epochs for both datasets.

Similar to the link prediction task, the relation prediction task aims to forecast the target relation between the head entity h and tail entity t. Specifically, for a given triple (h, r, t), we formulate a query for the relation prediction task: $(h, ?, t)$. Subsequently, each relation in the dataset is considered a potential target. We proceed to rank all candidate triples in descending order of score to determine the ranking of the ground-truth triple (h, r, t).

5.2. Evaluation metrics

This paper employs Mean Reciprocal Rank (MRR) and HIT Ratio with cut-off values of K (HIT@K) as our evaluation metrics. Higher MRR, higher HIT@K implies the better performance of the model. The definitions of MRR and HIT@K are shown in Eq.(16) and Eq.(17).

$$\mathrm{MRR} = \frac{1}{|F|} \sum_{i=1}^{|F|} \frac{1}{\mathrm{rank}_i} \tag{16}$$

$$\text{HIT @ K} = \frac{1}{|F|}\sum_{i=1}^{|F|} \boldsymbol{I}\left[\text{rank}_i \leqslant K\right] \tag{17}$$

5.3. Transductive Relation Prediction

We conduct a comparative analysis of HiHo with other state-of-the-art transductive methods. (TransE [6], DRUM [22], QuatE [21], P-INT [24], KE-GCN [32], GGAE [29], ConvRot [20]) in the transductive relation prediction experiment. Table 2 reports the results on five datasets. Specifically, the MRR increases of HiHo against the second-ranking method are 0.052, 0.006, 0.015, 0.015, and 0.039 on the five datasets respectively; the HIT@1 increases of HiHo against the second-ranking method are 2.5%, 0.9%, 2.7%, 2.3%, and 3.4% on the five datasets respectively; the HIT@3 increases of HiHo against the second-ranking method are 6.0%, 0.1%, 0.8%, 1.3%, and 0.3% on the five datasets respectively.

Table 2
The performance of different models in the transductive relation prediction experiment

Dataset	NELL995			FB15K			FB15K-237			WN18			WN18RR		
Metric	MRR	HIT@1	HIT@3	MRR	HIT@1	HIT@3	MRR	HIT@1	HIT@3	MRR	HIT@1	HIT@3	MRR	HIT@1	HIT@3
TransE	0.841	0.781	0.889	0.962	0.94	0.982	0.966	0.946	0.984	0.971	0.955	0.984	0.784	0.669	0.87
DRUM	0.715	0.64	0.74	0.945	0.945	0.978	0.959	0.905	0.958	0.969	0.956	0.98	0.854	0.778	0.912
QuatE	0.752	0.706	0.783	0.983	0.972	0.991	0.974	0.958	0.988	0.981	0.975	0.983	0.823	0.767	0.852
P-INT	0.706	0.644	0.727	0.885	0.839	0.966	0.941	0.885	0.943	0.954	0.931	0.971	0.832	0.763	0.905
KE-GCN	0.824	0.795	0.844	0.938	0.927	0.989	0.937	0.934	0.971	0.951	0.948	0.978	0.923	0.916	0.961
GGAE	0.845	0.815	0.881	0.934	0.928	0.985	0.937	0.931	0.959	0.968	0.943	0.977	0.945	0.942	0.991
ConvRot	0.856	0.826	0.892	0.935	0.962	0.996	0.975	0.957	0.982	0.983	0.974	0.985	0.948	0.931	0.992
HiHo	**0.908**	**0.874**	**0.946**	**0.989**	**0.981**	**0.997**	**0.99**	**0.984**	**0.996**	**0.998**	**0.997**	**0.998**	**0.987**	**0.974**	**0.995**

According to the experimental results shown in Table 2, we compared the HiHo with the baselines, and the analysis results are as follows:

(1) Our method outperforms all baselines in all datasets, especially in NELL995 dataset, where our method has significant advantages. The reason is that NELL995 is sparser than other benchmark datasets, which leads to the overfitting of the baselines and limits the learning ability of the baselines. The improvement of our method in NELL995 is quite significant because our method models the hierarchical and homogeneous nature of entities' subgraphs, which makes our method easy to extract effective features of entities. In addition, our method has fewer parameters and can be applied to some knowledge graphs which are characterized by having relatively fewer connections or edges between entities compared to the total number of possible relationships, i.e., sparse KGs.

(2) As translation-based method, TransE [6] and ConvRot [20] models only the inherent characteristics of entities and relations, it ignores the multi-step neighborhood features of entities, which limits its performance in relation prediction. As the semantic matching methods, DRUM [22] and QuatE [21] reason about what kind of relation exists between two entities by mining the underlying semantic association rules between entities and relations. However, they are very sensitive to rule settings, and some missing relations cannot be inferred from any rule [23]. As a path-based method, P-INT [24] can extract rich features in the paths between entities to predict the relations between entities. However, not all entities have paths between them, which leads to unpredictable relationships between them. As the graph-based methods, KE-GCN [32] and GGAE [29] learns the aggregated representation of entities by collecting information in their neighborhoods. However, the final representations of the head and tail entities are independent of each other, and they may have no common ground and lack the head-tail interactions, so the relation between them cannot be well reasoned.

5.4. Inductive Relation Prediction

We compare HiHo with other state-of-the-art inductive methods (GraIL [14], TACT [16], PathCon [38], StATIK [39], MorsE [37], RMPI [40]) in the inductive relation prediction experiment. Similar to the inductive KG completion experiment conducted by PathCon [38], we randomly sample approximately 20% of the entities present

in the test set. Subsequently, we exclude these entities along with their connected relations from the training set. The rest of the training set is utilized to train the models. During evaluation, we reintroduce the removed relations back into consideration [38]. As can be seen from the results in Figure 3 to Figure 7. Specifically, compared with other baselines, the MRR of HiHo are improved by 0.008∼0.184, 0.001∼0.056, 0.019∼0.066, 0.006∼0.042, and 0.006∼0.189 on the five datasets respectively; the HIT@1 of HiHo are improved by 0.5%∼21.0%, 0.5%∼6.2%, 1.2%∼4.9%, 0.2%∼6.5%, and 1.6%∼32.7% on the five datasets respectively; the HIT@3 of HiHo are improved by 0.6%∼25.4%, 0.2%∼2.2%, 0.1%∼3.1%, 0.1%∼1.9% and 0.2%∼13.1% on the five datasets respectively.

HiHo outperforms all other inductive methods. The reasons are: (1) GraIL, TACT and RMPI focus on modeling the intersection of two subgraphs, but sometimes the intersection only accounts for a very small proportion of the two subgraphs, which leads to many relation types in the subgraphs being ignored. (2) StATIK and MorsE model the subgraph of the head entity and the subgraph of the tail entity separately without considering the commonality of the two subgraphs. (3) PathCon predicts relations by alternately generalizing the context of two entities and modeling the relational path between two entities, but the relational path used by PathCon suffers from the same problems as the path-based methods. In addition, the above method only learns the features of the i-th layer subgraph of an entity separately, without modeling the hierarchical nature of the subgraph. Our method achieves the best performance by generalizing the hierarchical and homogeneous nature of the subgraphs of entities.

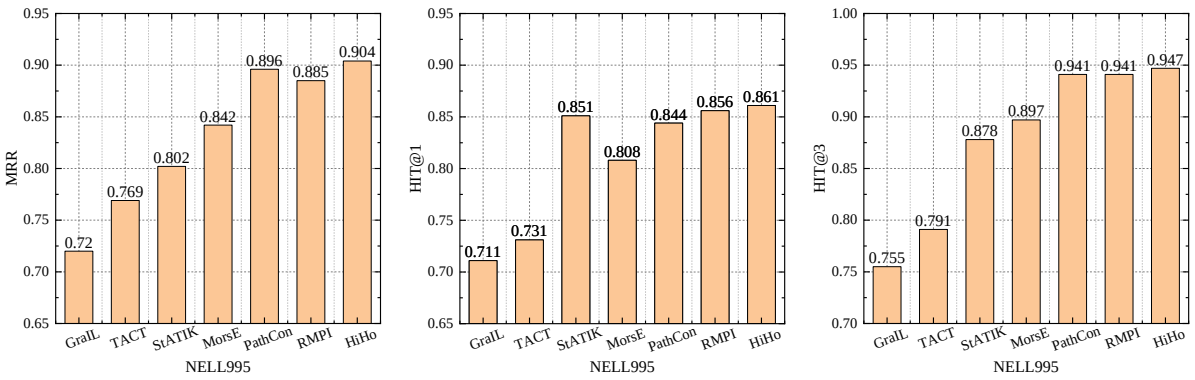

Fig. 3. The performance of each method on the NELL995 dataset.

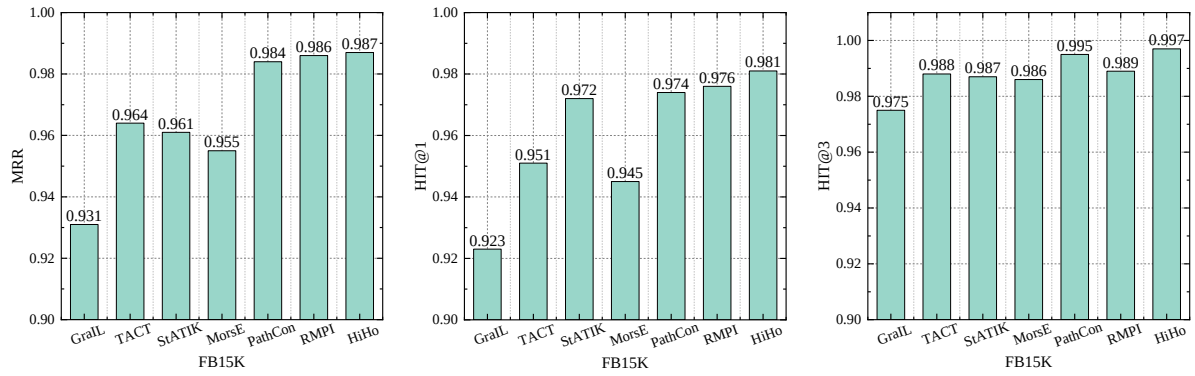

Fig. 4. The performance of each method on the FB15K dataset.

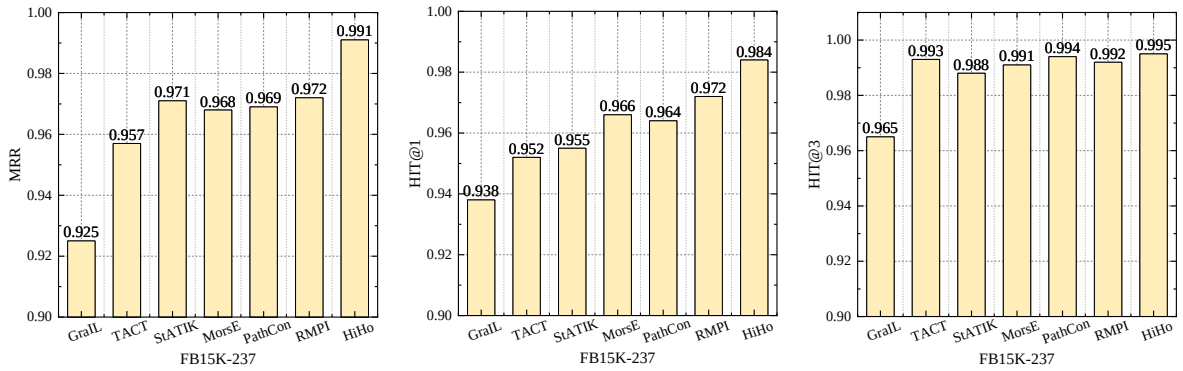

Fig. 5. The performance of each method on the FB15K237 dataset.

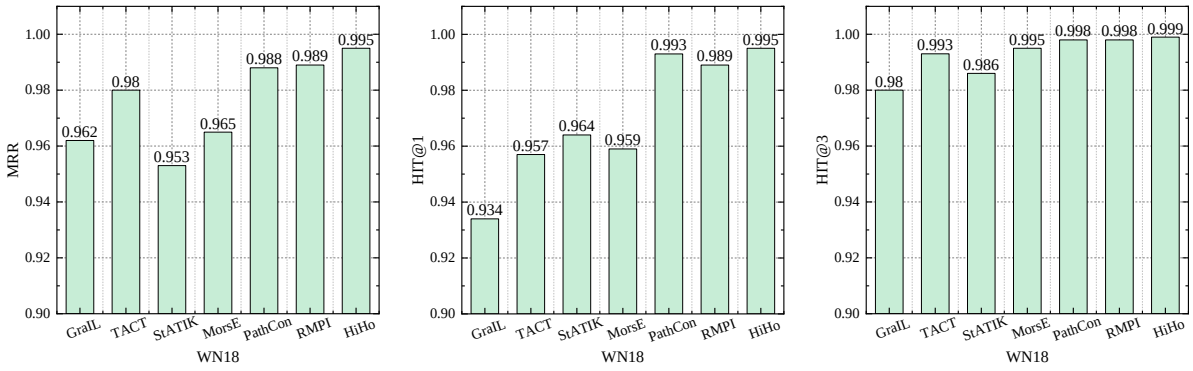

Fig. 6. The performance of each method on the WN18 dataset.

5.5. Ablation study

To demonstrate the individual effects of the proposed S2S and CPI components, we conduct an ablation experiment on the NELL995 dataset. Specifically, we exclude each component from the proposed method separately and evaluate their impact on the experimental results.

Table 3
Ablation study on NELL995 dataset

	MRR	HIT@1	HIT@3
Without S2S	0.862	0.791	0.907
Without CPI	0.885	0.812	0.931
HiHo	0.908	0.874	0.946

As shown in Table 3, compared to our whole model, the MRR, Hit@1, and Hit@3 of the model without S2S dropped by 0.23, 6.2%, and 1.5%, respectively; and the MRR, Hit@1, and Hit@3 of the model without CPI dropped by 0.46, 8.3%, and 3.9%, respectively. Therefore, we can obtain the following conclusions: (1) Without S2S, the performance of our model degrades severely, which proves that the lack of modeling the hierarchy of subgraph can seriously affect the effectiveness of relation prediction. (2) Without CPI, the performance of our model degrades slightly. The reason is that all the benchmark datasets used in this paper, even the NELL995 dataset, are relatively

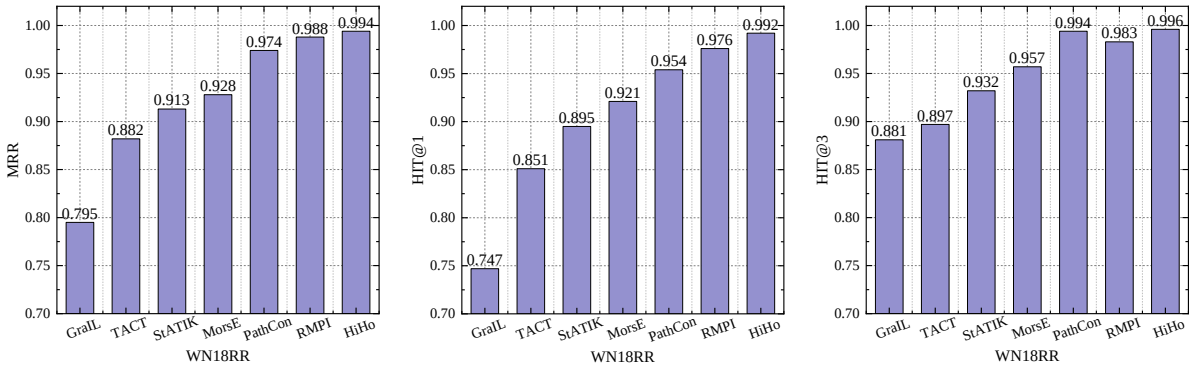

Fig. 7. The performance of each method on the WN18RR dataset.

dense.We quantify density as the ratio of existing edges to the total possible edges (calculated as $n(n-1)/2$ for a graph with n entities). The high density leads to dense relations in entities' subgraphs, different from some real-world KGs like DBpedia and Wikidata (not used in current experiments due to complexity and heterogeneity). In summary, the ablation study emphasizes the significance of S2S and CPI, especially S2S.

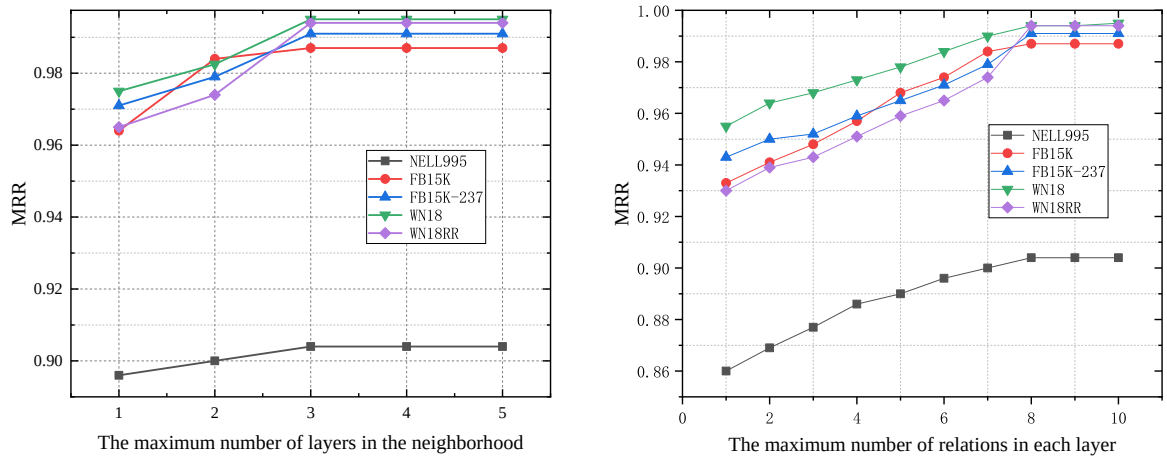

Fig. 8. The effect of the maximum number of layers in the neighborhood and the maximum number of relations in each layer

5.6. *Parameter sensitivity*

The maximum number of layers in the neighborhood L_{hop} , and the maximum number of relations in each layer N_r are the most important parameters of HiHo, and they largely determine the performance of HiHo in terms of KG relation prediction. In general, the larger the values of L_{hop} or N_r, the more valuable information HiHo can utilize to predict relations, thereby enabling a more comprehensive understanding of triple characteristics. However, excessively large values of L_{hop} or N_r can escalate the complexity of HiHo, diminish its training speed, and lead to the absorption of redundant information, ultimately resulting in overfitting. Conversely, setting L_{hop} or N_r too small may limit the features learned by HiHo, reducing its ability to predict relations effectively, resulting in underfitting.

In extreme cases, HiHo may regress to a translation-based or semantic matching method. Balancing these parameters is crucial for achieving optimal performance.

Fig. 8 shows the performance of the different values of L_{hop} and N_r on HiHo. From Fig. 8, we can see that: With the increase of L_{hop} and N_r, the MRR of HiHo increases continuously. When L_{hop} =3 and N_r =8, HiHo's MRR reaches the highest point and HiHo converges to the optimal solutions. But as L_{hop} and N_r continue to increase, the MRR of HiHo stops rising. Therefore, HiHo performs best on WN18 when L_{hop} =3 and N_r =8.

6. Conclusion

In this paper, we propose the HiHo model to predict the relation between entities in the inductive setting. Specifically, we propose a subgraph-to-sequence mechanism (S2S) to model the hierarchy of subgraph by learning the semantic associations between layers in the subgraph of a single entity, And we propose a common preference inference mechanism (CPI) to model the homogeneity of subgraph by assigning higher weights to co-occurrence relations and learning the importance of each relation in the subgraphs of two entities. Finally, we alternatively generalize the subgraphs of two entities to infer the relations between them. And compared with the baselines, our method has better performance.

Although the proposed method has shown good performance in the relation prediction task, it may encounter challenges when dealing with rare relations or complex subgraph structures. For rare relations, due to the lack of sufficient data, it is difficult to learn reliable patterns, which may lead to inaccurate predictions. In the future, we plan to adopt data augmentation techniques to alleviate this problem. For complex subgraph structures, the high connectivity and intertwined relationships may exceed the capacity of the model, making it difficult to capture meaningful information. In the future, we plan to develop a more advanced graph neural network architecture with hierarchical or multi-scale processing mechanisms to decompose the complexity and focus on relevant substructures, so as to achieve the effective extraction and utilization of information in complex subgraph structures. In addition, this paper has only explored the application of the proposed method in small and medium-sized real-world knowledge graphs. In future work, we will further explore its application in ultra-large-scale knowledge graphs.

Acknowledgements

This research was partly funded by National Natural Science Foundation of China(62272163), Henan Provincial Department of Science and Technology(242102211016), Songshan Laboratory Pre-research Project(YYJC012022023), Research and Practice Project of Research Teaching Reform in Undergraduate Colleges and Universities in Henan Province(202338870), Zhengzhou University of Light Industry(2021ZCKJ214, 2022BSJJ016), and Key Research Projects in Higher Education Institutions in Henan Province (25A520054).

References

[1] B. Shang, Y. Zhao, Y. Liu and C. Wang, Attention-based exploitation and exploration strategy for multi-hop knowledge graph reasoning, *Information Sciences* **653** (2024), 119787. doi:https://doi.org/10.1016/j.ins.2023.119787. https://www.sciencedirect.com/science/article/pii/S0020025523013725.

[2] J. Yang, X. Yang, R. Li, M. Luo, S. Jiang, Y. Zhang and D. Wang, BERT and hierarchical cross attention-based question answering over bridge inspection knowledge graph, *Expert Systems with Applications* **233** (2023), 120896. doi:https://doi.org/10.1016/j.eswa.2023.120896. https://www.sciencedirect.com/science/article/pii/S0957417423013982.

[3] F. Wang, Z. Zheng, Y. Zhang, Y. Li, K. Yang and C. Zhu, To see further: Knowledge graph-aware deep graph convolutional network for recommender systems, *Information Sciences* **647** (2023), 119465. doi:https://doi.org/10.1016/j.ins.2023.119465. https://www.sciencedirect.com/science/article/pii/S0020025523010502.

[4] S. Wu, X. Shen and R. Xia, Commonsense Knowledge Graph Completion Via Contrastive Pretraining and Node Clustering, in: *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J.L. Boyd-Graber and N. Okazaki, eds, Toronto, Canada, 2023, pp. 13977–13989. doi:10.18653/V1/2023.FINDINGS-ACL.878.

[5] B. Shi, H. Wang, Y. Li and S. Deng, RelaGraph: Improving embedding on small-scale sparse knowledge graphs by neighborhood relations, *Information Processing & Management* **60**(5) (2023), 103447. doi:https://doi.org/10.1016/j.ipm.2023.103447. https://www.sciencedirect.com/science/article/pii/S030645732300184X.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: *Advances in Neural Information Processing Systems*, Vol. 26, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani and K.Q. Weinberger, eds, Curran Associates, Inc., 2013. https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[7] Y. Lin, Z. Liu, M. Sun, Y. Liu and X. Zhu, Learning Entity and Relation Embeddings for Knowledge Graph Completion, *Proceedings of the AAAI Conference on Artificial Intelligence* **29**(1) (2015). doi:10.1609/aaai.v29i1.9491. https://ojs.aaai.org/index.php/AAAI/article/view/9491.

[8] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier and G. Bouchard, Complex Embeddings for Simple Link Prediction, in: *Proceedings of The 33rd International Conference on Machine Learning*, M.F. Balcan and K.Q. Weinberger, eds, Proceedings of Machine Learning Research, Vol. 48, PMLR, New York, New York, USA, 2016, pp. 2071–2080. https://proceedings.mlr.press/v48/trouillon16.html.

[9] B. Yang, W. Yih, X. He, J. Gao and L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2015. https://api.semanticscholar.org/CorpusID:2768038.

[10] D. Zhang, Z. Yuan, H. Liu, X. lin and H. Xiong, Learning to Walk with Dual Agents for Knowledge Graph Reasoning, *Proceedings of the AAAI Conference on Artificial Intelligence* **36**(5) (2022), 5932–5941. doi:10.1609/aaai.v36i5.20538. https://ojs.aaai.org/index.php/AAAI/article/view/20538.

[11] C. Fu, T. Chen, M. Qu, W. Jin and X. Ren, Collaborative Policy Learning for Open Knowledge Graph Reasoning, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng and X. Wan, eds, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2672–2681, https://doi.org/10.18653/v1/D19-1269.

[12] Y. Liu, S. Yang, J. Ding, quanming yao and Y. Li, Two Birds, One Stone: An Equivalent Transformation for Hyper-relational Knowledge Graph Modeling, 2023. https://openreview.net/forum?id=e3U6bGsfcA.

[13] D.Q. Nguyen, V. Tong, D. Phung and D.Q. Nguyen, Node Co-Occurrence Based Graph Neural Networks for Knowledge Graph Link Prediction, in *WSDM '22*, Association for Computing Machinery, New York, NY, USA, 2022, pp. 1589–1592–. ISBN 9781450391320. doi:10.1145/3488560.3502183.

[14] K. Teru, E. Denis and W. Hamilton, Inductive Relation Prediction by Subgraph Reasoning, in: *Proceedings of the 37th International Conference on Machine Learning*, H.D. III and A. Singh, eds, Proceedings of Machine Learning Research, Vol. 119, PMLR, 2020, pp. 9448–9457. https://proceedings.mlr.press/v119/teru20a.html.

[15] X. Xu, P. Zhang, Y. He, C. Chao and C. Yan, Subgraph Neighboring Relations Infomax for Inductive Link Prediction on Knowledge Graphs, in: *International Joint Conference on Artificial Intelligence*, 2022. https://api.semanticscholar.org/CorpusID:250629390.

[16] J. Chen, H. He, F. Wu and J. Wang, Topology-Aware Correlations Between Relations for Inductive Link Prediction in Knowledge Graphs, *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(7) (2021), 6271–6278. doi:10.1609/aaai.v35i7.16779. https://ojs.aaai.org/index.php/AAAI/article/view/16779.

[17] Z. Li, Q. Zhang, F. Zhu, D. Li, C. Zheng and Y. Zhang, Knowledge graph representation learning with simplifying hierarchical feature propagation, *Information Processing & Management* **60**(4) (2023), 103348. doi:https://doi.org/10.1016/j.ipm.2023.103348. https://www.sciencedirect.com/science/article/pii/S0306457323000857.

[18] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge Graph Embedding by Translating on Hyperplanes, *Proceedings of the AAAI Conference on Artificial Intelligence* **28**(1) (2014). doi:10.1609/aaai.v28i1.8870. https://ojs.aaai.org/index.php/AAAI/article/view/8870.

[19] A. Pavlović and E. Sallinger, ExpressivE: A Spatio-Functional Embedding For Knowledge Graph Completion, in: *The Eleventh International Conference on Learning Representations*, 2023. https://openreview.net/forum?id=xkev3_np08z.

[20] T. Le, N. Le and B. Le, Knowledge graph embedding by relational rotation and complex convolution for link prediction, *Expert Systems with Applications* **214** (2023), 119122. doi:https://doi.org/10.1016/j.eswa.2022.119122. https://www.sciencedirect.com/science/article/pii/S0957417422021406.

[21] S. ZHANG, Y. Tay, L. Yao and Q. Liu, Quaternion Knowledge Graph Embeddings, in: *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper_files/paper/2019/file/d961e9f236177d65d21100592edb0769-Paper.pdf.

[22] A. Sadeghian, M. Armandpour, P. Ding and D.Z. Wang, DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs, in: *Advances in Neural Information Processing Systems*, Vol. 32, 2019. https://proceedings.neurips.cc/paper_files/paper/2019/file/0c72cb7ee1512f800abe27823a792d03-Paper.pdf.

[23] Q. Lin, J. Liu, Y. Pan, L. Zhang, X. Hu and J. Ma, Rule-enhanced iterative complementation for knowledge graph reasoning, *Information Sciences* **575** (2021), 66–79.

[24] J. Xu, J. Zhang, X. Ke, Y. Dong, H. Chen, C. Li and Y. Liu, P-INT: A Path-based Interaction Model for Few-shot Knowledge Graph Completion, in: *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia and S.W.-t. Yih, eds, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 385–394. doi:10.18653/v1/2021.findings-emnlp.35.

[25] Z. Zhu, X. Yuan, M. Galkin, S. Xhonneux, M. Zhang, M. Gazeau and J. Tang, A*Net: A Scalable Path-based Reasoning Approach for Knowledge Graphs, 2022. https://api.semanticscholar.org/CorpusID:256597814.

[26] T. Ma, S. Lv, L. Huang and S. Hu, HiAM: A Hierarchical Attention based Model for knowledge graph multi-hop reasoning, *Neural Networks* **143** (2021), 261–270. doi:https://doi.org/10.1016/j.neunet.2021.06.008. https://www.sciencedirect.com/science/article/pii/S0893608021002409.
[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, Graph Attention Networks, in: *International Conference on Learning Representations*, 2018. https://openreview.net/forum?id=rJXMpikCZ.
[28] Y. Zhao, H. Feng, H. Zhou, Y. Yang, X. Chen, R. Xie, F. Zhuang and Q. Li, EIGAT: Incorporating global information in local attention for knowledge representation learning, *Knowledge-Based Systems* **237** (2022), 107909. doi:https://doi.org/10.1016/j.knosys.2021.107909. https://www.sciencedirect.com/science/article/pii/S0950705121010649.
[29] Q. Li, D. Wang, S. Feng, C. Niu and Y. Zhang, Global Graph Attention Embedding Network for Relation Prediction in Knowledge Graphs, *IEEE Transactions on Neural Networks and Learning Systems* **33**(11) (2022), 6712–6725. doi:10.1109/TNNLS.2021.3083259.
[30] T.N. Kipf and M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: *International Conference on Learning Representations*, 2017. https://openreview.net/forum?id=SJU4ayYgl.
[31] S. Vashishth, S. Sanyal, V. Nitin and P. Talukdar, Composition-based Multi-Relational Graph Convolutional Networks, in: *International Conference on Learning Representations*, 2020. https://openreview.net/forum?id=BylA_C4tPr.
[32] D. Yu, Y. Yang, R. Zhang and Y. Wu, Knowledge Embedding Based Graph Convolutional Network, in *WWW '21*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1619–1628–. ISBN 9781450383127. doi:10.1145/3442381.3449925.
[33] J. Zhang, J. Huang, J. Gao, R. Han and C. Zhou, Knowledge graph embedding by logical-default attention graph convolution neural network for link prediction, *Information Sciences* **593** (2022), 201–215. doi:https://doi.org/10.1016/j.ins.2022.01.076. https://www.sciencedirect.com/science/article/pii/S0020025522001153.
[34] J. Wang, Q. Zhang, F. Shi, D. Li, Y. Cai, J. Wang, B. Li, X. Wang, Z. Zhang and C. Zheng, Knowledge graph embedding model with attention-based high-low level features interaction convolutional network, *Information Processing & Management* **60**(4) (2023), 103350. doi:https://doi.org/10.1016/j.ipm.2023.103350. https://www.sciencedirect.com/science/article/pii/S0306457323000870.
[35] F. Yang, Z. Yang and W.W. Cohen, Differentiable Learning of Logical Rules for Knowledge Base Reasoning, in: *Neural Information Processing Systems*, 2017. https://api.semanticscholar.org/CorpusID:4502993.
[36] C. Wang, X. Zhou, S. Pan, L. Dong, Z. Song and Y. Sha, Exploring Relational Semantics for Inductive Knowledge Graph Completion, *Proceedings of the AAAI Conference on Artificial Intelligence* **36**(4) (2022), 4184–4192. doi:10.1609/aaai.v36i4.20337. https://ojs.aaai.org/index.php/AAAI/article/view/20337.
[37] M. Chen, W. Zhang, Y. Zhu, H. Zhou, Z. Yuan, C. Xu and H. Chen, Meta-Knowledge Transfer for Inductive Knowledge Graph Embedding, in: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 927–937–. ISBN 9781450387323. doi:10.1145/3477495.3531757.
[38] H. Wang, H. Ren and J. Leskovec, Relational Message Passing for Knowledge Graph Completion, in: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1697–1707–. ISBN 9781450383325. doi:10.1145/3447548.3467247.
[39] E. Markowitz, K. Balasubramanian, M. Mirtaheri, M. Annavaram, A. Galstyan and G. Ver Steeg, StATIK: Structure and Text for Inductive Knowledge Graph Completion, in: *Findings of the Association for Computational Linguistics: NAACL 2022*, M. Carpuat, M.-C. de Marneffe and I.V. Meza Ruiz, eds, Association for Computational Linguistics, Seattle, United States, 2022, pp. 604–615. doi:10.18653/v1/2022.findings-naacl.46.
[40] Y. Geng, J. Chen, W. Zhang, J.Z. Pan, M. Chen, H.-z. Chen and S. Jiang, Relational Message Passing for Fully Inductive Knowledge Graph Completion, *2023 IEEE 39th International Conference on Data Engineering (ICDE)* (2023), 1221–1233. https://api.semanticscholar.org/CorpusID:252780623.
[41] W. Xiong, T. Hoang and W.Y. Wang, DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa and S. Riedel, eds, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 564–573. doi:10.18653/v1/D17-1060.
[42] K. Xu, L. Wu, Z. Wang, Y. Feng, M. Witbrock and V. Sheinin, Graph2Seq: Graph to Sequence Learning with Attention-Based Neural Networks, 2019. https://openreview.net/forum?id=SkeXehR9t7.
[43] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge, in *SIGMOD '08*, Association for Computing Machinery, New York, NY, USA, 2008, pp. 1247–1250–. ISBN 9781605581026. doi:10.1145/1376616.1376746.
[44] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves and J. Welling, Never-Ending Learning, *Commun. ACM* **61**(5) (2018), 103–115–. doi:10.1145/3191513.
[45] G.A. Miller, WordNet: A Lexical Database for English **38**(11) (1995), 39–41–. doi:10.1145/219717.219748.
[46] D.P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2015.
[47] H. Hanke and D. Knees, A phase-field damage model based on evolving microstructure, *Asymptotic Analysis* **101** (2017), 149–180.
[48] E. Lefever, A hybrid approach to domain-independent taxonomy learning, *Applied Ontology* **11**(3) (2016), 255–278.
[49] P.S. Meltzer, A. Kallioniemi and J.M. Trent, Chromosome alterations in human solid tumors, in: *The Genetic Basis of Human Cancer*, B. Vogelstein and K.W. Kinzler, eds, McGraw-Hill, New York, 2002, pp. 93–113.
[50] P.R. Murray, K.S. Rosenthal, G.S. Kobayashi and M.A. Pfaller, *Medical Microbiology*, 4th edn, Mosby, St. Louis, 2002.
[51] E. Wilson, Active vibration analysis of thin-walled beams, PhD thesis, University of Virginia, 1991.

[52] C. Fu, T. Chen, M. Qu, W. Jin and X. Ren, Collaborative policy learning for open knowledge graph reasoning, *arXiv preprint arXiv:1909.00230* (2019).

[53] G. Ji, S. He, L. Xu, K. Liu and J. Zhao, Knowledge Graph Embedding via Dynamic Mapping Matrix, in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong and M. Strube, eds, Association for Computational Linguistics, Beijing, China, 2015, pp. 687–696. doi:10.3115/v1/P15-1067. https://aclanthology.org/P15-1067.

[54] P. Wang, J. Han, C. Li and R. Pan, Logic Attention Based Neighborhood Aggregation for Inductive Knowledge Graph Embedding **33** (2019), 7152–7159. doi:10.1609/aaai.v33i01.33017152. https://ojs.aaai.org/index.php/AAAI/article/view/4698.

[55] S. Liu, B. Grau, I. Horrocks and E. Kostylev, INDIGO: GNN-Based Inductive Knowledge Graph Completion Using Pair-Wise Encoding, in: *Advances in Neural Information Processing Systems*, Vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang and J.W. Vaughan, eds, Curran Associates, Inc., 2021, pp. 2034–2045. https://proceedings.neurips.cc/paper_files/paper/2021/file/0fd600c953cde8121262e322ef09f70e-Paper.pdf.