A Symbolic Rule Integration Framework with Logic Transformer for Inductive Relation Prediction

Yudai Pan†
School of Computer Science and Technology, Xi’an Jiaotong University
Xi’an, Shaanxi, China
pyd418@foxmail.com

Jun Liu∗
National Engineering Lab for Big Data Analytics, Xi’an Jiaotong University
Xi’an, Shaanxi, China
liukeen@xjtu.edu.cn

Tianzhe Zhao
School of Computer Science and Technology, Xi’an Jiaotong University
Xi’an, Shaanxi, China
ztz8758@foxmail.com

Lingling Zhang
Shaanxi Provincial Key Laboratory of Big Data Knowledge Engineering, Xi’an Jiaotong University
Xi’an, Shaanxi, China
zhanglling@xjtu.edu.cn

Yun Lin†
Shanghai Jiao Tong University
Shanghai, China
lin_yun@sjtu.edu.cn

Jin Song Dong
National University of Singapore
Singapore
dcsdjs@nus.edu.sg

ABSTRACT

Relation prediction in knowledge graphs (KGs) aims at predicting missing relations in incomplete triples, whereas the dominant paradigm by KG embeddings has a limitation to predict the relation between unseen entities. This situation is called an inductive setting, which is more common in the real-world scenario. To handle this issue, implicit symbolic rules have shown great potential in capturing the inductive capability. However, it is still challenging to obtain precise representations of logic rules from KGs. The argument variability and predicate non-commutativity in symbolic rule integration make the modeling of component symbols difficult. To this end, we propose a novel inductive relation prediction model named SymRITa with a logic transformer integrating rules. SymRITa firstly extracts the subgraph, whose embeddings are captured by a graph network. Meanwhile, symbolic rule graphs in the subgraph can be generated. Then, the symbolic rules are modeled by a proposed logic transformer. Specifically, the input format based on the subgraph-based embeddings is to focus on the argument variability in symbolic rules. In addition, a conjunction attention mechanism in the logic transformer can resolve predicate non-commutativity in the symbolic rule integration process. Finally, the subgraph-based and symbol-based embeddings obtained from the previous steps are combined for the training regime, and prediction results as well as rules explaining the reasoning process are explicitly output. Extensive experiments on twelve inductive datasets show that SymRITa achieves outstanding effectiveness compared to state-of-the-art inductive baselines. Moreover, the logic rules with corresponding confidences provide an interpretable paradigm.

CCS CONCEPTS

• Computing methodologies → Knowledge representation and reasoning.

KEYWORDS

Inductive relation prediction; knowledge graph; first-order logic; logic transformer

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:

1 INTRODUCTION

In a structured storage scenario, knowledge graphs (KGs) possess numerous triples consisting of entities and relations, representing factual knowledge about the real world. They have been widely used in various downstream tasks, such as question answering [1, 11], information retrieval [6, 35], and text generation [16], etc. Existing KGs suffer from incompleteness according to the open-world assumption, which weakens the performance of downstream tasks. Therefore, some dominant methods aim at predicting missing entities and relations using the rich implicit structural information. There are many conventional methods for obtaining the relation and entity embeddings, such as TransE [3], R-GCN [29], CompGCN [34] and StAR [36].

Previous embedding-based methods are merely designed for a transductive setting. However, in the real-world scenario, the relation prediction is always implemented in an inductive setting, which predicts the relation between two unseen entities in the test set. The above-mentioned methods are not suitable for this situation, because they require to retrain the whole model for unseen entities and lack an inductive ability in applications. With the development of large language models (LLMs), some recent methods try to implement reasoning by LLMs [46, 51], while they require massive manual prompts and the performance does not meet the desired outcome, especially on unseen out-of-distribution samples.
To address the above issues, we propose a Symbolic Rule Integration model with logic Transformer for inductive relation prediction named SymRITa. In SymRITa, the logic transformer module is proposed to obtain embeddings of symbolic rules during reasoning. In detail, we firstly extract the enclosing subgraph among the target triple, which performs as the foundation to extract implicit rules. Following the format of the first-order logic rule, we extract them within a maximum preset length as the skeletons. Secondly, we implement the subgraph modeling and obtain the subgraph-based embeddings of entities and relations through a graph network based on the graph topological information. Then, we propose a rule integration strategy to integrate the implicit logic and obtain symbol-based embeddings for inductive ability. The inputs of logic transformer are from the subgraph-based relation embeddings, which aim at the argument variability in the integration. As for the predicate non-commutativity, a conjunction attention providing undamaged symbolic information mechanism is designed for the logic transformer, providing comprehensive information for symbolic rule integration. Finally, SymRITa implements the inductive relation prediction by combining subgraph-based and symbol-based embeddings and jointly trains the model. The symbol-based embedding is obtained by each symbolic rule and its corresponding confidence, which illustrates the explanation of the reasoning process.

Our main contribution are in the following three folds:

- A novel framework named SymRITa integrates the symbolic rules into the continuous neural network model to solve the inductive relation prediction. To the best of our knowledge, it is the first method to model symbol-based embeddings and combine them with subgraph-based embeddings for obtaining the inductive ability.
- In order to model the discrete symbolic rules for resolving entity independence in the inductive setting, we propose a logic transformer and design a conjunction attention mechanism to satisfy the argument variability and strengthen the predicate non-commutativity of first-order logic rules extracted from KGs.
- Experiments of the relation prediction on twelve inductive datasets verify the effectiveness of SymRITa compared to the latest inductive methods. Meanwhile, SymRITa obtains first-order logic rules with confidences, which can potentially explain the reasoning process.

2 RELATED WORK

2.1 Inductive Relation Prediction

For the inductive relation prediction task on KGs, we divide them into two main categories: rule-based and graph-based.

Rule-based methods extract logic rules from KGs enhancing the inductive ability instead of fusing with external information [39, 40]. Associated methods mine closed-paths rules from enumerated candidates in KGs by statistical indicators, such as head coverage in AMIE [13, 14], pointwise mutual information in SHERLOCK [30], and other indicators in early works [21, 25]. RuleN [22] extends AMIE with a fine-grained evaluation, which extracts more rules with high qualities. Considering the lack of scalability of these associated methods, some other differential methods solve
the inductive problem by learnable weights for rules during the reasoning process in KGs. NeuralLP [45] learns the confidences and the structures of logic rules through an end-to-end model, which consists of a controller system implemented by a recurrent neural network (RNN). DRUM [28] extends NeuralLP with the Bi-RNN to extract logic rules with flexible length. Both of the methods are based on the framework named TensorLog [7], which transfers the discrete rule extracting to continuous learnable parameters. Nevertheless, these methods are still struggling with inadequate inductive reasoning performance based on the framework with limited inductive ability and high resource consumption.

Graph-based methods try to take account of the topological information of KGs to get more sufficient inductive ability. GraIL [32] mines the inductive ability of KGs from extracted subgraphs and models them by a graph neural network (GNN). CoMPLEX [20] is proposed based on GraIL considering the implicit inductive ability by interacting messages between edges and entities. TACT [5] extends GraIL by a relational correlation network to supply different patterns of topological information between relations. RED-GNN [52] models the KG by a relational directed graph, in order to capture the local evidence of the KG for inductive reasoning. However, these methods merely utilize the topological information without taking advantage of the implicit semantics.

Distinguished from the previous methods, SymRITa combines the topological structure with symbolic rules, which is significant for improving the inductive ability of the model.

2.2 Symbolic Reasoning
Symbolic reasoning aims to integrate symbolic formulas into the reasoning tasks, which would not only develop the performance but also the interpretability of the reasoning process. For the visual problems, NS-VQA [47] incorporates symbolic structure as prior knowledge to a visual question answering task. It recovers a structural scene representation from the image and a symbolic program trace from the question, and then executes the program to obtain an answer. For the logical question answering task, LReasoner [37] proposes a context extension framework based on logical equivalence laws to capture symbolic logic from the text. As for the graph networks, SGR [19] performs reasoning over a group of symbolic nodes whose outputs explicitly represent different properties of semantics in a prior graph. It learns shared symbolic representations for domains or datasets with the different label set.

Symbolic reasoning is applied to text [37, 42], graphs [19], images and multi-modal scenarios [47]. For solving the inductive setting and getting the inductive capability, we innovatively introduce the symbolic reasoning into the inductive relation prediction.

3 PRELIMINARY

3.1 Inductive Relation Prediction
Inductive relation prediction in KGs aims to predict the relation between two unseen entities. A target triple is denoted as \((h, r, t)\) in the train KG \(G = \{R, E, T\}\), in which \(r \in R\) is the target relation. \(h, t \in E\) are head and tail entities, respectively. \(R, E\) and \(T\) are sets of relations and entities in \(G\), and \(T \subseteq E \times R \times E\) is the set of triples. Inductive relation prediction intends to predict if the triple \((h', r, t')\) is valid with two unseen entities \(h'\) and \(t'\) in a testing KG \(G' = \{R, E', T'\}\). \(G\) and \(G'\) share the same relation set. However, entities in \(G\) and \(G'\) are disjoint, i.e. \(E' \cap E = \emptyset\). For clarity, we summarize important symbols in Table 4 in Appendix B.

3.2 Symbolic Rule in KGs
The symbolic rule [23] learned from KGs is a first-order logic Horn rule [27], which consists of a head atom and a series of atoms as the body. Here is a first-order logic rule with length \(N\):

\[
\text{head} \quad \beta r_1(X, Y) \iff r_1(X, Z_1) \land r_2(Z_1, Z_2) \land \cdots r_n(Z_{N-1}, Y). \tag{1}
\]

where \(r_1, r_2, \ldots, r_N\) are predicates, represented as relations in KGs. \(X, Z_1, Z_2, \ldots, Z_{N-1}, Y\) are arguments generalized from entities in KGs. The body atoms \(r_1(X, Z_1), \ldots, r_n(Z_{N-1}, Y)\) are connected by a conjunction symbol \(\land\) and point to the head by an implication symbol \(\iff\). In rule (1), an atom contains two arguments and adjacent atoms share the same argument. During inference, arguments are instantiated to entities \(x, z_1, z_2, \ldots, z_{N-1}, y\) and form a closed path. In the KG, we can use the target relation to simulate the head predicate of the rule. Also, \(\beta \in [0, 1]\) can be used to represent the confidence of the rule. According to [45], these are important components in the first-order logic rules extracted from KGs.

4 METHODOLOGY
In this section, we demonstrate SymRITa with the help of the framework in Figure 2. SymRITa is in three parts.

4.1 Subgraph Modeling
SymRITa extracts the enclosing subgraph \(S_T\) based on the target triple \((h, r, t)\) from \(G\). The entities in \(S_T\) are selected by the intersection of \(q\)-hop undirected neighborhoods of the target triple. The triples among the selected entities conduct the subgraph \(S_T\). Furthermore, \(S_T\) is used for labeling nodes by a double radius vertex labeling scheme [50]. The labeling process initializes the entities in the subgraph, and provides the initialized vector with the inductive ability by the topological structure [32]. Then, we initialize the node \(i\) with a vector \([\text{one-hot}(d(i, h)) \oplus \text{one-hot}(d(i, t))] \in \mathbb{R}^{(2q+2)}\) as the node feature, where \(\oplus\) is the concatenation operation of vectors. \(d(\cdot)\) generates the shortest topological distance between two entities.

For subgraph modeling, we employ a graph neural network (GNN) [29] based approach with relation-aware and triple-aware attention to obtain entity and relation embeddings in a \(S_T\). In detail, the update process of the entity embeddings is:

\[
\alpha^{(l+1)} = \phi\left( \sum_{r \in R, j \in N_r^i} \alpha^{(l)} \beta_{i,r} \mathbf{W}_r a^{(l)}_j + \mathbf{W}_0 \alpha^{(l)}_i \right), \tag{2}
\]

\[
\omega^{(l)}_{i,j,r} = \sigma_1(W_1^{(l)}[a^{(l)}_i \oplus a^{(l)}_j \oplus r]), \tag{3}
\]

\[
\alpha^{(l)}_{i,j,r} = \sigma_2(W_2^{(l)}[\omega^{(l)}_{i,j,r} \oplus r]), \tag{4}
\]

where we use \(\alpha^{(l+1)}\) to be the embedding of node \(i\) in \((l+1)\)-th layer from \(L_1\) layers during message passing. \(\phi\) denotes the multilayer perceptron (MLP) operation to aggregate the neighbor information. \(N_r^i\) collects the neighbors of node \(i\) by relation \(r\). \(\mathbf{W}_r^{(l)}\) and \(\mathbf{W}_0^{(l)}\) are transformation matrices to update the message from layer \(l\) to
As for the symbol components in rules, we design a strategy to handle them for enhancing the inductive ability during reasoning. This is the main part of our SymRITa, and we propose a graph transformer-based method to model the rules in subgraphs.

4.2 Symbolic Rule Integration

As for the symbol components in rules, we design a strategy to handle them for enhancing the inductive ability during reasoning. This is the main part of our SymRITa, and we propose a graph transformer-based method to model the rules in subgraphs.

4.2.1 Logic Extraction and Sorter. We construct the implicit logic rules from the extracted subgraphs. From the format of the first-order logic rule as shown in Eq. (1), the relational paths in a subgraph are significant parts of the implicit logic rules in a KG. If we get the relational paths from the head to tail of the target triple \( r_T \), we could obtain the skeletons of logic rules in \( S_T \). Specifically, we use the breadth first search (BFS) algorithm [8] for extracting every topological relational path whose length is not longer than \( L_{\text{max}} \) from \( h \) to \( t \) in \( S_T \). The set of \( n \) logic rules among \( a_T \) is denoted as \( \{ f_1, \ldots, f_n \} \). If \( L_{\text{max}} \) is preset as 3, \( S_T \) owns 4 relational paths from Figure 2 to represent the implicit logic rules for reasoning the relation between \( h \) to \( t \). According to Eq. (1), the arguments (e.g. \( X, Y, Z \)) and conjunction symbols (i.e. \( \land \)) are significant in rules, so the predicates in a single extract relational path should be in order, for instance, the relation order of \( f_1 = \{ r_1, r_2 \} \) in \( S_T \). The information of arguments and the semantics of conjunction symbol in the rules is fused in SymRITa, which will be demonstrated in the following sections.

Then, the symbolic rules are sorted by their semantics in order to match the significance of the corresponding rule during reasoning. From [10], the order of inputs is significant in the transformer. It is inspired by the encoder of the language model, that adjacent tokens have similar positional embeddings. Therefore, we calculate the similarity of semantics and make the adjacent rules have similar semantics. Based on a rule learning work [44], we design a strategy to calculate the semantic similarity between \( r_T \) and the rule body \( f_k \), in which \( r_T \) and \( f_k = \{ r_{k,1}, \ldots, r_{k,n} \} \) connect the same \( h \) and \( t \).\( n \) is the number of relations in the format of \( f_k \). In SymRITa, we denote the similarity to be \( \theta_k \):

\[
\theta_k = \varphi(f_k) \cdot r_T = (\sum_{r_i \in f_k} r_i)^\top r_T
\]

where \( \varphi \) is an aggregator for combining all the relations in \( f_k \). The value of the semantic similarity \( \theta_k \) indicates the rationality of a first-order logic rule body \( f_k \). We sort them by values of \( \theta \) in a descending order, and represent them as a sequence \( \mathcal{F} = \{ f_1', f_2', \ldots, f_n' \} \), in which \( n \) indicates the number of bodies in \( \mathcal{F} \). This process can determine the relation sequence in symbolic rule integration, which is a reference for the positional information sent to the next part named logic transformer.

4.2.2 Symbolic Rule Graph and Logic Transformer. As a critical part of the rule integration, we propose a module to embed not only the semantics of relations in rules, but also the semantics of arguments and conjunction symbols. Inspired by the work of graph transformer [49], the architecture of logic transformer is devised. From the previous subsection, SymRITa treats the relations to be the predicates in the first-order rules. Because of the entity independence, we should not embed the exact entities in \( S_T \). Therefore, we design a module aiming at different arguments in the adjacent atoms connected by conjunction symbols.

As described in Section 3, when considering the arguments, they are always fused with the semantics of predicates to construct the body atoms in a first-order logic rule. For example, the head atom consists of the target relation \( r_T \) and exact arguments \( X \) and \( Y \), while
the body relations of the rule are connected by conjunctions \( \land \) and arguments with variability. It can be structured to a directed graph, which is shown in the gray block in Figure 2 named symbolic rule graph. The nodes refer to atoms in the body (gray nodes), and the directed edges (blue arrows) are conjunction symbols connecting the same argument in adjacent atoms. Directed symbolic rule graphs follow the perspective of the reasoning process by the first-order logic rules as Eq. (1).

The input of the logic transformer for each \( S_T \) consists of the relation embeddings from elements in \( F \), which satisfies the requirement of the model for argument variability:

\[
F = \{ r_{1,1}; r_{1,2}; \cdots; r_{1,n_1}; \cdots; r_{n,1}; r_{n,2}; \cdots; r_{n,n_2} \}. \tag{6}
\]

According to the sorting result of the sorter, the position number of a node is determined by its position in the body \( f'_k \) and the position of \( f'_k \) in \( F \). Generally, we number the relations in \( F \) and consider them to be the position information of the logic transformer. Therefore, the positional embeddings are added into the rule embeddings, which are denoted as:

\[
F_o = F + \text{PosEmb}(F) \tag{7}
\]

where \( \text{PosEmb}(\cdot) \) provides a positional embedding to each node in the input from directed graphs. A more detailed illustration of \( \text{PosEmb}(\cdot) \) is in Appendix A.1.

Afterwards, we feed the overall node embedding \( F_o \) into the logic transformer. As shown in Figure 3, the self-attention module is implemented by three matrices \( Q, K, V \), which indicate the query, key and value in the transformer:

\[
Q^i, K^i, V^i = F_o W^j_q, F_o W^j_k, F_o W^j_v, \tag{8}
\]

where \( W^j_q, W^j_k \) and \( W^j_v \) are trainable parameter matrices of the \( i \)-th head in the logic transformer, which project the input \( F_o \) to the corresponding representations \( Q^i, K^i \) and \( V^i \). Then, the attention is calculated by the matrices and the conjunction attention:

\[
\text{Attn}^i(F_o) = \text{softmax}(A^i) V^i, \tag{10}
\]

in which \( M_{\text{conj}} \) is the conjunction attention we propose to introduce the operator information to the rule integration. \( \lambda \) is a hyper-parameter adjusting the weight of the conjunction attention. \( d_k \) is the scaling factor. We use the adjacent matrix \( M_{\text{conj}} \) to represent the symbolic rule graph. The element in \( M_{\text{conj}} \) is defined as \( M_{\text{conj}}[i, j] = 1 \) when the nodes \( i \) and \( j \) have a directed edge or a path. For example, in Figure 3, the position of the nodes are from the symbolic graphs corresponding to the input of the logic transformer, which is represented as Eq. (6). For the rule body \( r_5(X, Z_1) \land r_1(Z_2, Y) \), the elements of \( M_{\text{conj}} [4, 5], M_{\text{conj}} [4, 6] \) and \( M_{\text{conj}} [5, 6] \) are set to 1. These three elements in \( M_{\text{conj}} \) simultaneously represent the directed graph in Figure 3.

After \( L_2 \) layers, the representations of the hidden states \( H(L_2) \) are treated as the output of the logic transformer:

\[
H(L_2) = [h_{1,1}; h_{1,2}; \cdots; h_{1,n_1}; \cdots; h_{n,1}; h_{n,2}; \cdots; h_{n,n_2}]. \tag{11}
\]

### 4.3 Training and Prediction

After modeling the topological and logical information by subgraphs and symbolic rules, we fuse the representations and use them to train the model and implement the reasoning process. Specifically, the overall subgraph-based embedding \( S \) is denoted as:

\[
\begin{aligned}
\sigma_T^{(L_1)} &= \frac{1}{|V_T|} \sum_{i \in V_T} \sigma_T^{(L_1)}_i, \tag{12} \\
S &= \sigma_T^{(L_1)} @ v_{\text{arg}}^{(L_1)} @ r_T, \tag{13}
\end{aligned}
\]

in which \( \sigma_T^{(L_1)} \) indicates the concatenated embeddings of entities. \( V_T \) is the set of nodes in \( S_T \). In this process, we apply the JK-connection [32, 43] in getting the subgraph-based embedding \( \sigma_T^{(L_2)} = \left[ \bigoplus_{L=1}^{L_2} ( \sigma_T^{(L_1)} \oplus \sigma_T^{(L_1)} ) \right] \).

For the symbol-based embedding, we use the output \( H(L_2) \) of the logic transformer to obtain the overall symbolic rule representation \( P \) in \( S_T \) and the confidence \( \beta_k \) of symbolic rule \( f'_k \):

\[
P = \sum_{k=1}^{n} \beta_k f'_k = \sum_{k=1}^{n} \beta_k \sum_{j=1}^{l_k} h_{k,j}, \tag{14}
\]

\[
\beta_k = \text{softmax}(f'_k, r_T) = \frac{\exp(f'_k \top r_T)}{\sum_{f'_p \in F} \exp(f'_p \top r_T)}, \tag{15}
\]

where \( n \) is the number of rules within length \( L_{\text{max}} \) in \( S_T \). Eventually, we evaluate the target triple by the link prediction, so the score of \( a_T \) combining subgraph and symbolic information is denoted as:

\[
\text{score}(a_T) = W_{\text{score}} [S @ P], \tag{16}
\]

where \( W_{\text{score}} \) refers to the weight matrix. Then the training process is implemented by a margin-based loss [38] to distance scores of positive and negative samples:

\[
\mathcal{L} = \sum_{a_T \in \mathcal{E}} \max(0, \eta + \text{score}(a_T^+) - \text{score}(a_T^-)), \tag{17}
\]

in which \( \mathcal{E} \) refers to the set of triples in \( G \), and \( \eta \) is the margin representing the distance. \( a_T^+ \) indicates the positive sample and \( a_T^- \) refers to the negative one. As for symbolic rules, we randomly replace predicates in \( S_T \) to construct negative samples.
5 EXPERIMENTAL RESULTS

In this section, we firstly introduce datasets, baselines, experiment settings, and details. Then, we implement comparison experiments, ablation studies, weight analysis and others to demonstrate the performance of SymRITa.

5.1 Datasets and Baselines

Datasets. We follow the inductive link prediction in the basic model GraIL [32] which proposes benchmarks derived from WN18RR [9], FB15K-237 [33] and NELL-995 [41], and each has been divided into four versions. Each version of a dataset consists of a pair of KGs named train-graph and ind-graph, whose entities are without intersection. Each train-graph has its valid KG for evaluation during training. The statistics of benchmark datasets are illustrated in Table 5 in the Appendix C.1.

Baselines. For comparison, we select typical models for the inductive relation prediction task, including rule-based RuleN [22], Neural-LP [45] and DRUM [28], and graph-based GraIL [32], CoMPiLE [20], TACT [5], and recent works RED-GNN [52], LogCo [26], and RMPI [15]. For a fair comparison, all the results of baselines are from published papers, which are implemented by a consistent link prediction setting.

5.2 Metrics and Experimental Settings

Metrics. In the comparison and other tasks for evaluating the performance of SymRITa, we implement both classification and ranking metrics for multiple runs considering the random seeds and samples. For the classification task, AUC-PR is an indicator to calculate the area under the precision-recall curve and it evaluates the ranking of each positive triple in the test set. For the ranking metric Hits@k, we evaluate it in a general link prediction mode by ranking the score of test triples among 50 random negative samples, and see if the true triple can rank in the top k.

Experimental Details. For the subgraph extraction, we obtain 3-hop enclosing subgraphs by the double vertex labeling. In the graph embedding process, we employ a 3-layer GNN with the dimension of 64. For the logic transformer, we set the dimension of the hidden state as 256, the number of heads as 6, and the number of layers as 2. For the hyper-parameters λ, we set it to 0.8 and illustrate the reason in Section 5.5. During the training process, we set the batch size as 16 and we use Adam [18] as the optimizer with a learning rate being 0.0005. We set the margin value in the training loss is \( \lambda \), we set it to 0.0005. The maximum length of rules is set as \( m_{\text{max}} = 2, 3 \) and the margin value in the training loss is 5. We implement the experiments on one NVIDIA’s Tesla V100 graphic card. More detailed settings are in Appendix C.2.

5.3 Comparison Results

In this subsection, we use the two metrics on the classification and ranking tasks respectively to illustrate the effectiveness of SymRITa. SymRITa is evaluated on twelve inductive benchmarks of KGs from [32]. The reasoning results are shown in Table 1. All the results of baselines are from published papers at top conferences.

For the classification task reflected by the metric AUC-PR, SymRITa can obviously outperform all the selected state-of-the-art baselines, since it obtains optimal or suboptimal values of AUC-PR on all the datasets in Table 1. Specifically, except for WN18RR, SymRITa outperforms all the listed inductive relation prediction methods on the classification task. Even on WN18RR, SymRITa gets the suboptimal value of AUC-PR, which is 0.74% slightly lower than the SOTA method CoMPiLE. Nevertheless, the average boost of SymRITa on WN18RR compared to CoMPiLE is 0.28%. As for...
Table 2: Ablation results on inductive benchmarks derived from FB15K-237_v1 and NELL-995_v1.

<table>
<thead>
<tr>
<th>Method</th>
<th>FB15K-237_v1</th>
<th>NELL-995_v1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR H@1 H@5 H@10 AUC-PR</td>
<td>MRR H@1 H@5 H@10 AUC-PR</td>
</tr>
<tr>
<td>SymRITa</td>
<td>51.98 42.43 62.20 72.20 89.77</td>
<td>50.34 44.00 56.00 62.50 92.59</td>
</tr>
<tr>
<td>SymRITa w/o Symbolic Rules</td>
<td>48.68 39.75 57.07 65.61 86.01</td>
<td>47.56 42.70 48.80 54.20 83.53</td>
</tr>
<tr>
<td>Δ</td>
<td>[2.30] [2.68] [5.13] [6.59] [13.76]</td>
<td>[2.78] [1.30] [7.20] [8.30] [9.06]</td>
</tr>
<tr>
<td>SymRITa w/o Logic Transformer</td>
<td>48.56 35.54 58.04 66.34 87.87</td>
<td>48.11 42.50 53.50 58.50 85.27</td>
</tr>
<tr>
<td>Δ</td>
<td>[3.42] [3.89] [4.16] [5.86] [11.90]</td>
<td>[2.23] [1.50] [2.50] [4.00] [7.32]</td>
</tr>
<tr>
<td>SymRITa w/o Conjunction Attention</td>
<td>49.77 40.00 60.98 69.26 88.04</td>
<td>49.92 43.50 53.00 60.10 87.85</td>
</tr>
<tr>
<td>Δ</td>
<td>[2.21] [2.43] [1.22] [2.94] [1.73]</td>
<td>[0.42] [0.50] [3.00] [2.40] [4.74]</td>
</tr>
</tbody>
</table>

The ablation results in Table 2 illustrate the effectiveness of the significant factors in SymRITa. (1) **The inductive performance achieves optimal results when all components work simultaneously.** In both FB15K-237_v1 and NELL-995_v1, the integration of three factors helps SymRITa obtain obvious classification and ranking improvements. (2) **The extracted rules from the subgraph are essential in inductive relation prediction.** From the AUC-PR and link prediction results of SymRITa w/o Symbolic Rules, the reductions indicate the significance of incorporated rules in the model. Specifically, compared to the original SymRITa, SymRITa w/o Symbolic Rules reduces the prediction results by 4.29% and 5.73% on two datasets respectively. The main reason is that the implicit logics contained in the rules help improve the inductive ability. (3) **The logic transformer and conjunction attention in it are also critical in SymRITa.** The prediction performance of SymRITa w/o Logic Transformer averages drops by 3.85% and 3.51% on FB15K-237_v1 and NELL-995_v1, respectively. The reduction of SymRITa w/o Conjunction Attention is slighter, but the average decline compared to SymRITa still illustrates its effectiveness.

5.4 Ablation Studies

We investigate the impacts of rules and the logic transformer in obtaining the inductive ability of the model. As shown in Table 2, we rerun the model without the factors on FB15K-237_v1 and NELL-995_v1 respectively, and the methods are denoted as:

- **SymRITa w/o Symbolic Rules** indicates the method removing the constructed rules in the subgraph.
- **SymRITa w/o Logic Transformer** still considers the semantics of rules, which are represented by relational paths neglecting the arguments and conjunctions. To implement this model, we replace the logic transformer with the mean operator. It can be regarded as the effectiveness of the logic transformer and the conjunction attention simultaneously.
- **SymRITa w/o Conjunction Attention** indicates the method removing the conjunction attention in the logic transformer.

The ablation results in Table 2 illustrate the effectiveness of the significant factors in SymRITa. (1) **The inductive performance achieves optimal results when all components work simultaneously.** In both FB15K-237_v1 and NELL-995_v1, the integration of three factors helps SymRITa obtain obvious classification and ranking improvements. (2) **The extracted rules from the subgraph are essential in inductive relation prediction.** From the AUC-PR and link prediction results of SymRITa w/o Symbolic Rules, the reductions indicate the significance of incorporated rules in the model. Specifically, compared to the original SymRITa, SymRITa w/o Symbolic Rules reduces the prediction results by 4.29% and 5.73% on two datasets respectively. The main reason is that the implicit logics contained in the rules help improve the inductive ability. (3) **The logic transformer and conjunction attention in it are also critical in SymRITa.** The prediction performance of SymRITa w/o Logic Transformer averages drops by 3.85% and 3.51% on FB15K-237_v1 and NELL-995_v1, respectively. The reduction of SymRITa w/o Conjunction Attention is slighter, but the average decline compared to SymRITa still illustrates its effectiveness.

5.5 Weight Analysis

Analysis of the sorting mode. During training, the sorting mode of the sorter is critical in the rule integration process. It decides the order of rules in the same subgraph. We record the classification and ranking results with an ascending sorter and a descending sorter respectively on FB15K-237_v1. The results are shown in Figure 4(a). It can be figured that the descending order benefits both classification and ranking results with an ascending sorter and a descending sorter respectively on FB15K-237_v1. The results on FB15K-237_v1 are in Figure 4(b). From the results of classification metric AUC-PR and ranking metric Hits@1, when η = 5, SymRITa obtains the most effective performance. It indicates that a large or small margin will lead to a negative impact.

Analysis of the margin η. In SymRITa, we use a parameter η in the loss function to optimize the training process. In the loss function, it decides the gap between the positive and negative samples. The results on FB15K-237_v1 are in Figure 4(b). From the results of classification metric AUC-PR and ranking metric Hits@1, when η = 5, SymRITa obtains the most effective performance. It indicates that a large or small margin will lead to a negative impact.

Analysis of hyper-parameter λ. In SymRITa, λ controls the conjunction attention weight in the logic transformer when modeling the rules. We rerun the training process on FB15K-237_v1 and NELL-995_v1, and record the results of classification and ranking tasks in different values of λ ∈ [0.1, 1.2] with a step of 0.1. The results of AUC-PR and MRR are shown in Figure 5. From the distributions of results on FB15K-237_v1, we observe that the inductive
As we illustrate in Section 3, the rule from KGs is in a form of the first-order logic with its corresponding confidence \( \beta \). In Table 3, it illustrates the first-order logic rules with the confidences of SymRITa and the model without the essential factor logic transformer on NELL-995_v1. SymRITa can adjust the importance of each rule by obtaining precise embeddings of rules in a subgraph. For instance, if the target relation is \( \text{agent} \_\text{controls} \), the two rules in the same subgraph are shown in the last row of Table 3. The logic transformer increases the significance of rule \( \text{agent} \_\text{controls} \leftarrow \text{company} \_\text{economic} \_\text{sector}(X,Z) \land \text{agent} \_\text{controls}(Z,Y) \), which is reflected by the improvement of \( \beta \) from 0.1579 to 1. In reality, we can figure that the rule \( \text{agent} \_\text{controls}(X,Y) \leftarrow \text{subpart} \_\text{of} \_\text{organization}(X,Z) \land \text{agent} \_\text{controls}(Z,Y) \) will not be reasonable in reasoning the query \( (h, \text{agent} \_\text{controls}, t) \) where \( \text{agent} \_\text{controls} \) is the target relation. In most situation, the subpart could not represent the entire organization, which would lead to an incorrect reasoning result.

Figure 5: Effectiveness evaluation by MRR (%) and AUC-PR(%) of parameters \( \lambda \) over two datasets.

### 6 Conclusion and Future Work

In order to solve the entity independence of inductive relation prediction in KGs, we propose a method named SymRITa to integrate symbolic rules during the reasoning process. First, we extract subgraphs based on the target triple, and rules from each subgraph. Second, we obtain the subgraph-based embeddings by a GNN and symbol-based embeddings by the logic transformer with a conjunction attention mechanism. The logic transformer is designed for the challenges of argument variability and predicate non-commutativity in modeling rules and preserving their precise representations. Finally, SymRITa captures the inductive ability by combining subgraph-based and symbol-based embeddings, and jointly trains the model. The experiments on twelve inductive datasets show the effectiveness of SymRITa, and comprehensively demonstrate the impacts of rules and the significance of factors from our model. SymRITa still needs improving in terms of performance and scalability. We will extend the symbolic rules whose atoms are connected by disjunction, and expand the inductive scenario to commonsense knowledge graphs [31]. Also, we will extend the logic transformer to rule extraction and embedding decoupling in larger train sets, fusing not only closed-path rules but also rules with flexible forms.

### Acknowledgments

This work was supported by National Key Research and Development Program of China (2022YFC3303600), National Natural Science Foundation of China (62137002, 62293550, 62293553, 62293554, 62176207, 62106190, 62192781, and 62250009), Innovative Research Group of the National Natural Science Foundation of China (61721002), “LENOVO-XJTU” Intelligent Industry Joint Laboratory Project, Foundation of Key National Defense Science and Technology Laboratory (6142101210201), National Nature Science Foundation of China (61721002). The Youth Innovation Centre for Engineering Science and Technology, XJTU Teaching Reform Research Project “Acquisition Learning Based on Knowledge Forest”, the Ministry of Education, Singapore (T2EP2012-0019, MOET32020-0004), and National Research Foundation, Singapore, and the Cyber Security Agency under its National Cybersecurity R&D Programme (NCRP25-P04-TAICeN), DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023-008).
A SUPPLEMENT OF SYMRITA

A.1 Illustration of Positional Embedding

In Equation 7, SymRITA uses $\text{PosEmb}()$ to obtain the positional embedding to each node in the input from directed graphs. For the $2i$ and $2i+1$-th elements ($i = 0, 1, \ldots, d/2 - 1$, $d$ is an even number) of each vector in $F$, $\text{PosEmb}(\cdot)$ is indicated as:

$$\text{PosEmb}(\text{pos}, 2i) = \sin(\text{pos}/10000^{2i/d})$$

in which $\text{pos}$ is the corresponding position of the vector in $F$, and $d$ is the dimension of $r$ in $F$.

A.2 Relation Embedding in Subgraph Modeling

As for the relation embedding in GNN in Equation 2, it is a random initialized embedding, and updated with each iteration [29]. It is represented by the updating of $\mathbf{W}_r$, which is regularizing of the weight with different relation types:

$$\mathbf{W}_r^{(l)} = a_r^{(l)} \mathbf{V}^{(l)}$$

which is a linear combination of basis transformations $\mathbf{V}^{(l)}$ with coefficients $a_r^{(l)}$ such that only the coefficients depend on $r$ in the $l$-th layer.

B TRAINING PROCEDURE

SymRITA can predict missing relations of incomplete triples in an inductive setting, and also integrates first-order rules for explanation at the same time. In Algorithm 1, we demonstrate the process of predicting relations by SymRITA. SymRITA uses a KG as the input and at last outputs the score of target triple and a set of first-order rules with confidences. For clarity, we summarize important symbols in Table 4.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_T$</td>
<td>Target triple to be predicted</td>
</tr>
<tr>
<td>$G, G^r$</td>
<td>Train and test KGs for inductive prediction</td>
</tr>
<tr>
<td>$\mathbf{S}_r$</td>
<td>The subgraph of $a_T$</td>
</tr>
<tr>
<td>$\mathbf{V}^{(L_2)}$</td>
<td>Embedding vector of node $i$ at layer $L_1$</td>
</tr>
<tr>
<td>$\mathbf{L}_{\text{max}}$</td>
<td>Max length of the relational paths</td>
</tr>
<tr>
<td>$\mathbf{F} = {f_1, f_2, \ldots, f_n}$</td>
<td>Symbolic rules extracted from $\mathbf{S}_r$</td>
</tr>
<tr>
<td>$\mathbf{P}$</td>
<td>Output of logic transformer after $L_2$ layers</td>
</tr>
</tbody>
</table>

C EXPERIMENTAL DETAILS

C.1 Datasets

The statistics of benchmark datasets are illustrated in Table 5. These datasets consist of train and test KGs with the inductive setting. Each version of a dataset consists of a pair of KGs named train-graph and ind-test-graph, whose entities are without intersection. Meanwhile, train-graph contains all the relations in ind-test-graph. Each train-graph has its valid KG for evaluation during training.

We also evaluated the inductive reasoning ability of SymRITA in a more challenging scenarios, where unseen relations also occur during testing. We implement experiments on datasets NELL-995.v1.v3, NELL-995.v4.v3 and FB15k-237.v1.v4, released in [15]. The detailed statistics of these datasets are listed in Table 6, and the results are discussed in Appendix D.4.
Algorithm 1 Process of integrating rules by SymRITa

Input: KG $G (R, E, T)$, target triple $a_T$, hyper-parameters $\lambda, \eta, \text{etc.}$

Output: Score of $a_T$ and First-order logic rules.

1. Extract subgraph $S_T$ around each $a_T$, and initialize each node.
2. for each training iteration do
3. for each batch of triples in $G$ do
4. Obtain embeddings of entities and relations in $S_T$.
5. Extract rules within the length $L_{\text{max}}$ and generate symbolic rule graphs.
6. $F \leftarrow$ Sort the rules with values of $\theta$.
7. $F \leftarrow$ Get input of the logic transformer.
8. $F_0 \leftarrow$ Add position information PosEmb$(F)$ in the symbolic rule graphs.
9. $M_{\text{conj}} \leftarrow$ Obtain the matrix in the conjunction attention by symbolic rule graphs.
10. Calculate self-attention with matrices $Q, K, V$ and $M_{\text{conj}}$ by Eq. (10) and Eq. (9).
11. $H^{(L)} \leftarrow$ Output the logic transformer.
12. Get subgraph-based embedding $S$ and symbol embedding $P$ with confidence of each rule $\beta$.
13. Score $a_T$ combining $S$ and $P$.
14. $L \leftarrow$ Get margin-based loss by Eq. (17).
15. Update the parameters by Adam optimizer.
16. end for
17. end for
18. return Score of $a_T$ and first-order rules in $F$ with $\beta$.

Table 5: Statistics of Inductive Relation Prediction Datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>WN18RR</th>
<th>FB15k-237</th>
<th>NELL-995</th>
</tr>
</thead>
<tbody>
<tr>
<td>#R</td>
<td>#E</td>
<td>#S</td>
<td>#T</td>
</tr>
<tr>
<td>v1</td>
<td>9</td>
<td>2,746</td>
<td>6,687</td>
</tr>
<tr>
<td>v2</td>
<td>10</td>
<td>6,954</td>
<td>18,968</td>
</tr>
<tr>
<td>v3</td>
<td>11</td>
<td>12,078</td>
<td>32,150</td>
</tr>
<tr>
<td>v4</td>
<td>14</td>
<td>2,923</td>
<td>7,470</td>
</tr>
<tr>
<td>v1-ind</td>
<td>9</td>
<td>922</td>
<td>1,991</td>
</tr>
<tr>
<td>v2-ind</td>
<td>10</td>
<td>1,991</td>
<td>3,980</td>
</tr>
<tr>
<td>v3-ind</td>
<td>11</td>
<td>3,861</td>
<td>6,678</td>
</tr>
<tr>
<td>v4-ind</td>
<td>14</td>
<td>7,088</td>
<td>18,968</td>
</tr>
</tbody>
</table>

Table 6: Statistics of datasets with unseen relations in the test set. The numbers in the brackets are the numbers of unseen relations.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NELL-995 v1-v3</th>
<th>NELL-995 v4-v3</th>
<th>FB15k-237 v1-v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>#R</td>
<td>#E</td>
<td>#S</td>
<td>#T</td>
</tr>
<tr>
<td>train</td>
<td>14</td>
<td>3305</td>
<td>5440</td>
</tr>
<tr>
<td>test</td>
<td>106 (98)</td>
<td>2271</td>
<td>5550</td>
</tr>
</tbody>
</table>

C.2 Experimental Settings

Different parameters might influence the performance of the logic transformer\(^3\) on different datasets, so the parameters are tuned separately. We conduct experiments in the following search space of parameters:

- Learning rate: [0.0001, 0.0002, 0.0005, 0.001]
- The number of subgraph hops: {2, 3, 4}
- Maximum length of relational paths $L_{\text{max}}$: \{1, 2, 3\}
- Margin hyper-parameter $\eta$: \{3, 4, 5, 6\}
- Number of heads in logic transformer: \{5, 6, 7, 8\}
- Number of layers of GNN $L_1$: \{2, 3, 4\}
- Number of layers in logic transformer $L_2$: \{2, 3, 4\}
- $d_k, d_q$ and $d_p$ [48]: \{32, 64, 128\}

D SUPPLEMENTARY RESULTS

D.1 Results on Other Metrics

In Section 5, Hits@10 represents a general result for ranking task. Since the papers of baselines only record the values of Hits@10 of all the 12 datasets, we rerun the baselines with open codes. We compare the values of Hits@1, 5, 10, mean reciprocal rank (MRR) and AUC-PR. Table 7 shows some results of Hits@1, 5, 10, MRR, AUC-PR on WN18RR_v1 compared to GraIL\(^4\), which is the basic GNN-based method. It gives more comprehensive results to indicate the effectiveness of SymRITa on more metrics.

Table 7: Reasoning results of ranking and classification tasks on different metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>MRR</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>AUC-PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraIL</td>
<td>79.57</td>
<td>73.79</td>
<td>84.04</td>
<td>84.04</td>
<td>94.32</td>
</tr>
<tr>
<td>SymRITa</td>
<td>80.48</td>
<td>77.65</td>
<td>86.97</td>
<td>91.22</td>
<td>99.58</td>
</tr>
</tbody>
</table>

D.2 Results of Different Sorting Mode

According to the previous work of transformer [10], the order of inputs is significant in the transformer. It is inspired by the encoder of the language model, that adjacent tokens have similar positional embeddings. Therefore, we calculate the similarity of semantics and make the adjacent rules have similar semantics. Table 8 shows some results with random order of rules comparing to the ascending and descending orders. It is shown that the ordered rules are more effective than rules with a random order. As for whether the rules are in ascending or descending order, we analyze the results in Section 5.5. The descending order obtains better inductive reasoning results, which may be because that the more important rule with its corresponding position information can precisely embed the symbols of rules in a subgraph.

Table 8: Reasoning results of different sorting modes of the sorter in the rule integration process. The sorting mode are random order, descending order and ascending order.

<table>
<thead>
<tr>
<th>Sorting Mode</th>
<th>MRR</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>AUC-PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descending</td>
<td>48.44</td>
<td>37.56</td>
<td>60.24</td>
<td>70.24</td>
<td>88.05</td>
</tr>
<tr>
<td>Ascending</td>
<td>44.77</td>
<td>34.39</td>
<td>57.56</td>
<td>65.61</td>
<td>84.36</td>
</tr>
<tr>
<td>Random</td>
<td>42.90</td>
<td>31.71</td>
<td>56.82</td>
<td>64.87</td>
<td>84.12</td>
</tr>
</tbody>
</table>

Besides, inspired by the work of rule learning [45], we use the recurrent neural network (RNN) to replace the sum operation in Eq. 5 and call it SymRITa-RNN. Then, the $\theta_k$ is calculate as $RNN(f_k)^\top r_T$. Table 9 shows some results of SymRITa-RNN on WN18RR_v1. It is

\(^3\)https://github.com/pyd418/LogicTransformer

\(^4\)https://github.com/kkteru/grail
shown that the results are not apparently influenced by the relation order within the rules. As shown in Table 8, the order of the relations in a rule will not seem to affect the semantic similarity, i.e. $\theta$, before sending them to the logic transformer. The order of rules is important for the performance. This phenomenon may be caused by that the relation embeddings do not contain the information of arguments and conjunctions, so roughly considering the order of relations when calculating $\theta$ may not have a significant impact.

Table 9: Reasoning results of different aggregator for combining the relations in sorting process.

<table>
<thead>
<tr>
<th>Aggregator</th>
<th>MRR</th>
<th>Hits@1</th>
<th>AUC-PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymRITa-sum</td>
<td>80.48</td>
<td>77.65</td>
<td>99.58</td>
</tr>
<tr>
<td>SymRITa-RNN</td>
<td>80.46</td>
<td>78.45</td>
<td>99.15</td>
</tr>
</tbody>
</table>

D.3 Results in Transductive Scenario

We have implemented an experiment to indicate the effectiveness of SymRITa in transductive scenario. The test entities are seen in the train set. The value of AUC-PR of SymRITa on WN18RR_v1 is 95.44% and AUC-PR of GraIL is 57.63%. The inductive reasoning is a more challenging task compared to transductive learning, and SymRITa can solve both, which illustrate the scalability of SymRITa in different reasoning scenarios.

D.4 Unseen Relation Prediction

Based on the training settings of SymRITa, the trained model can only be tested with semi unseen relations, so we implement the testing on three of the datasets to illustrate the effectiveness.

We record the ranking results of MRR on three test datasets that models by SymRITa can be tested, which are in Table 10. The ranking results outperform the baselines on three inductive datasets. The performance indicates that the symbolic rule integration helps SymRITa to deal with the issue with several unseen relations mixed in the test set.

Table 10: Ranking results of MRR (%) on inductive benchmarks with unseen relations derived from NELL-995.v1.v3, NELL-995.v4.v3 and FB15k-237.v1.v4

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TACT-base</td>
<td>43.59</td>
<td>52.68</td>
<td>61.02</td>
</tr>
<tr>
<td>RMPI-base</td>
<td>59.10</td>
<td>70.33</td>
<td>56.81</td>
</tr>
<tr>
<td>RMPI-NE</td>
<td>56.19</td>
<td>59.47</td>
<td>57.77</td>
</tr>
<tr>
<td>SymRITa</td>
<td>65.54</td>
<td>70.71</td>
<td>61.64</td>
</tr>
</tbody>
</table>