Rule-Guided Graph Neural Networks for Explainable Knowledge Graph Reasoning

Zhe Wang[1], **Suxue Ma**[2*], **Kewen Wang**[1], **Zhiqiang Zhuang**[2]
[1]Griffith University
[2]Tianjin University
zhe.wang@griffith.edu.au, msx@tju.edu.cn, k.wang@griffith.edu.au, zhuang@tju.edu.cn

Abstract

The connections between symbolic rules and neural networks have been explored in various directions, including rule mining through neural networks and rule-based explanation for neural networks. These approaches allow symbolic rules to be extracted from neural network models, which offers explainability to the models. However, the plausibility of the extracted rules is rarely analysed. In this paper, we show that the confidence degrees of extracted rules are generally not high, and we propose a new family of Graph Neural Networks that can be trained with the guidance of rules. Hence, the inference of our model simulates the rule reasoning. Moreover, rules with high confidence degrees can be extracted from the trained model that aligns with the inference of the model, which verifies the effectiveness of the rule guidance. Experimental evaluation of knowledge graph reasoning tasks further demonstrates the effectiveness of our model.

Introduction

Both neural networks and symbolic rules have been widely applied to perform inferences over structured data, such as knowledge graphs (KGs) (Hogan et al. 2021). Approaches based on various neural networks, especially Graph Neural Networks (GNNs), often demonstrate superior accuracy than rule-based approaches in KG inference tasks such as link prediction (Schlichtkrull et al. 2018; Teru, Denis, and Hamilton 2020; Yan et al. 2022), due to the capability of neural networks to model more complex inference patterns (beyond those captured by rule-based reasoning) through numeric manipulations. However, their reasoning processes are often hard to interpret. On the other hand, approaches based on symbolic reasoning have the clear advantage of explainability (Wu et al. 2023; Chen et al. 2024).

Recently, there has been an emerging interest in neuro-symbolic approaches by combining the inference power of neural networks with the explainability of symbolic rules. Several approaches have been proposed for end-to-end rule mining through neural networks (Yang, Yang, and Cohen 2017; Sadeghian et al. 2019; Qu et al. 2020; Cheng, Ahmed, and Sun 2022). These approaches often simulate rule reasoning using neural networks, which allows them to evaluate candidate rules and extract rules from the neural networks. Yet it is found that the extracted rules may not *faithfully* capture the inference of the neural networks; that is, the inference results of the models and the extracted rules may have discrepancies. In particular, the extracted rules are not necessarily *sound*, in the sense that some facts derived from the rules cannot be inferred from the corresponding model.

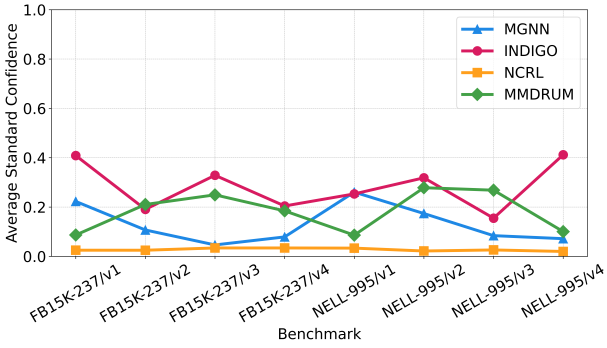

Figure 1: The average standard confidence of the rules extracted by explainable GNN-based models MGNN (Cucala et al. 2021) and INDIGO (Liu et al. 2021), and end-to-end rule learners NCRL (Cheng, Ahmed, and Sun 2022) and MMDRUM (Wang et al. 2023) on 8 benchmarks.

In order to better align neural network models with rules for explainability, research efforts have been made to design neural networks whose inference can be faithfully captured by rules (Cucala et al. 2021; Cucala, Grau, and Motik 2022; Wang et al. 2023). In particular, it is shown that for a GNN model $\mathcal{M}$ satisfying certain monotonic conditions, its inference results on KGs coincide with those of a finite set of Datalog rules $\mathcal{R}_\mathcal{M}$; that is, rules $\mathcal{R}_\mathcal{M}$ can be used to explain the inference of $\mathcal{M}$ (Cucala et al. 2021). Although these results provide a faithful and symbolic explanation of GNNs, the *quality* of the explanation was not sufficiently analysed. In the rule mining literature, the quality of rules is typically measured by their statistical confidence degrees (called *standard confidence*) over the KGs. As shown in Figure 1, the standard confidence degrees of the rules extracted by existing methods are relatively low. This suggests that the plausibility of the rules $\mathcal{R}_\mathcal{M}$, which refers to how be-

*Corresponding author.

lievable or reasonable the rules are in the context of domain knowledge or common sense, is not necessarily verified. For example, the rule $\mathsf{team\_also\_known\_as}(x, y) \leftarrow \mathsf{team\_plays\_against\_team}(y, x)$ extracted by the GNN proposed in (Cucala et al. 2021) on the NELL-995 dataset has a low confidence and does not match the KG.

Thus, it would be interesting and helpful to design explainable neural models $\mathcal{M}$ that possess the following desirable properties:

(**P1**) A finite set of rules $\mathcal{R}_\mathcal{M}$ can be efficiently extracted from $\mathcal{M}$.

(**P2**) Rules in $\mathcal{R}_\mathcal{M}$ are sound w.r.t. $\mathcal{M}$ for KG inference.

(**P3**) The confidence degrees of the rules in $\mathcal{R}_\mathcal{M}$ are high.

To the best of our knowledge, no existing explainable models satisfy (**P1**) – (**P3**), because the training of the models are not guided by rules. It is non-trivial to inject symbolic rules to guide neural network models, and the existing rule-injection approaches (Niu et al. 2020; Nayyeri et al. 2021; Liu et al. 2024; Pan et al. 2024) cannot be easily adapted to explainable neural models. Indeed, in these approaches, the relationship between the inference of the "guided" models and that of the injected rules is not clearly stated, while the effectiveness of rule guidance is often evaluated only through the performance of KG reasoning. Hence, we propose a fourth property on rule guidance.

(**P4**) The training of $\mathcal{M}$ can be effectively guided by a set of rules $\mathcal{R}$.

For an approach satisfying both (**P1**) and (**P4**), the effectiveness of the rule guidance can be evidenced by the correlation between $\mathcal{R}$ and $\mathcal{R}_\mathcal{M}$.

In this paper, we propose a new family of GNNs that satisfy all properties (**P1**) – (**P4**). We achieve this by establishing a correspondence between the parameters of GNNs and rules. Such a correspondence enables the GNNs can be effectively guided by rules. Moreover, it allows us to conveniently "read off" rules from the parameters of the GNNs, which are proven to be *sound* for the GNN inference. This provides a way to analyse the effectiveness of our rule guidance. Experimental evaluations show the effectiveness of our approach, by demonstrating superior performance in KG link prediction than other rule mining and explainable GNN models. Also, we show that the quality of the rules extracted from our model is outstanding.

Related Work

Our research is related to neuro-symbolic approaches to KG reasoning, including end-to-end rule mining via neural networks, rule-based explanations for neural networks, especially for GNNs, and rule-guided neural reasoning.

End-to-end rule mining approaches evaluate candidate rules via neural network models and extract rules from the models (Yang, Yang, and Cohen 2017; Sadeghian et al. 2019; Qu et al. 2020). While these approaches satisfy (**P1**) but they do not necessarily satisfy (**P2**). In fact, it is reported that the inference of the extracted rules may not coincide with the model (Cucala, Grau, and Motik 2022; Wang et al. 2023). Faithful versions of Neural-LP (Cucala, Grau, and Motik 2022) and DRUM (Wang et al. 2023) have been proposed, where the inference of the model and rules coincide. The performance of these faithful approaches is still inferior to that of GNN models like GraIL (Teru, Denis, and Hamilton 2020) and INDIGO (Liu et al. 2021). Also, the quality of the extracted rules of these approaches has not been well analysed, i.e., they do not necessarily satisfy (**P3**). In our approach, we use quality rules to guide the training of our model to ensure the quality of extracted rules.

Another stream of research proposes explainable GNNs through symbolic means (Cucala et al. 2021; Liu et al. 2021; Zhang et al. 2023). In particular, rules can be extracted from trained monotonic GNN models (Cucala et al. 2021, 2023), whose inference fully coincides with the models'. Formal results on the expressivity of such monotonic GNNs have been established through their correspondence with Datalog rules (Cucala et al. 2023). Similar to those end-to-end rule mining approaches, the quality of the rules extracted from the monotonic GNNs has not been sufficiently studied. That is, they do not necessarily satisfy (**P3**). Also, the rule extraction algorithms involve enumerating and evaluating all candidate rules, which are of high computational complexity even though several optimization techniques are provided. In this paper, we propose a new family of monotonic GNNs where rules can be "read off" from the GNN parameters.

Another group of research works related to ours are rule-guided neural models for KGs (Guo et al. 2016, 2018; Niu et al. 2020; Nayyeri et al. 2021; Liu et al. 2024; Pan et al. 2024). Yet, these approaches cannot be easily extended to explainable GNNs. In particular, they cannot provide symbolic explanations to the model, i.e., they do not satisfy (**P1**) – (**P3**). Indeed, the relationship between the inference of the models and that of the injected rules is not stated, while the effectiveness of rule guidance is often evaluated only through the performance of KG reasoning. In contrast, our approach allows rules to be extracted from the model to explain the inference of the model, and by comparing the injected and extracted rules, to evaluate to what extent the injected rules actually guide the model.

Preliminaries

We recall some basics of knowledge graphs, rules, and graph neural networks, as well as some notations used in the paper.

A *knowledge graph* (*KG*) is a set of *triples* of the form (s, p, o), where the *subject* s and the *object* o are associated via the *relation* p. In this paper, we distinguish triples of the form (e, type, c), which says an entity e has a type c, from other triples in the KG; for example, $(\texttt{LeBron}, \mathsf{type}, \texttt{Person})$ says `LeBron` has a type of `Person`. That is, we distinguish types from entities. We denote the sets of entities, types, and relations (other than type) as $\mathcal{E}$, $\mathcal{C}$, and $\mathcal{P}$ respectively. Relations and types can be seen as binary and unary predicates, and a KG can be considered a set of *facts* of the forms $p(e, e')$ and $c(e)$ with $e, e' \in \mathcal{E}$, $p \in \mathcal{P}$, and $c \in \mathcal{C}$. We also call a set of facts a *dataset*.

A (Horn) rule is of the form

$$H \leftarrow B_1 \wedge \cdots \wedge B_n, \tag{1}$$

where H and B_i's ($1 \leq i \leq n$) are *atoms* of the forms $p(x, y)$ and $c(x)$ with p being a binary predicate, c a unary predicate, and x, y variables. H is the *head* of the rule and B_i's ($1 \leq i \leq n$) form the *body* of the rule. The *length* of the rule is n, i.e., the number of body atoms. A *substitution* θ is a mapping from variables to entities. For an atom, a set of atoms, a rule, or a set of rules φ, $\theta(\varphi)$ is the result of replacing all the variables in φ with entities according to θ. For a dataset $\mathcal{D}$, the result of *applying* rule r on $\mathcal{D}$ is defined as the smallest set of facts $r(\mathcal{D})$ containing $\theta(H)$ for each substitution θ from the variables in r to the entities in $\mathcal{D}$ such that $\theta(B_1), \ldots, \theta(B_n) \in \mathcal{D}$. For a set of rules $\mathcal{R}$, $\mathcal{R}(\mathcal{D}) = \bigcup_{r \in \mathcal{R}} r(\mathcal{D})$.

Most existing rule learners for KGs use *standard confidence* to evaluate the statistical plausibility of rules (Galárraga et al. 2013; Omran, Wang, and Wang 2019). For a dataset $\mathcal{D}$ and a rule r of the form (1), let $\vec{x}$ be the variables in H, $h(r, \mathcal{D}) = \{\theta(\vec{x}) \mid \theta(H) \in \mathcal{D}\}$ and $b(r, \mathcal{D}) = \{\theta(\vec{x}) \mid \theta(B_1), \ldots, \theta(B_n) \in \mathcal{D}\}$. Then, the *standard confidence* (*SC*) of r over $\mathcal{D}$ is defined as follows

$$sc(r, \mathcal{D}) = \frac{|h(r, \mathcal{D}) \cap b(r, \mathcal{D})|}{|b(r, \mathcal{D})|}. \quad (2)$$

The higher the value is, the more plausible the rule is.

A *graph neural network* (*GNN*) (Scarselli et al. 2009) is a deep learning framework designed to process graph-structured data. It updates the features of each node in a graph by aggregating feature information from its neighbourhood. Formally, the representation of node v in the l-th GNN layer is given by

$$\mathbf{a}_v^{(l)} = \mathsf{AGGREGATE}^{(l)}(\{\mathbf{h}_u^{(l-1)} \ : \ u \in \mathcal{N}_v\}), \quad (3)$$

$$\mathbf{h}_v^{(l)} = \mathsf{COMBINE}^{(l)}(\mathbf{h}_v^{(l-1)}, \mathbf{a}_v^{(l)}), \quad (4)$$

where $\mathcal{N}_v$ denotes the neighbours of v, $\mathsf{AGGREGATE}^{(l)}$ is a function that combines the features of neighbouring nodes, and $\mathsf{COMBINE}^{(l)}$ integrates the aggregated neighbour information with the current node's $(l-1)$-th layer representation to produce its l-th layer representation. Initial node features $\mathbf{h}_v^{(0)}$ are typically derived from the node's attributes.

Our Approach

In this section, we present our approach to rule-guided GNNs for KG reasoning. Building on the pair encoding strategy in existing explainable GNN approaches (Cucala et al. 2021; Liu et al. 2021), we first transform the KG $\mathcal{K}$ into an inference graph $\mathcal{G}_\mathcal{K}$ to be processed by a GNN, with the results of KG inference being directly decoded from the numeric vectors associated with the nodes in $\mathcal{G}_\mathcal{K}$.

KG Encoding and Decoding

The nodes in the inference graph $\mathcal{G}_\mathcal{K}$ can be divided into a set $\mathcal{U}$ of *singleton* nodes and a set $\mathcal{V}$ of *pair* nodes. $\mathcal{U}$ consists of all the entities in the KG. For an entity e, let $\mathcal{N}_e^k$ denote all the entities in $\mathcal{K}$ that can be reached within k-hops ($k \geq 1$) disregarding the edge directions. Then, $\mathcal{V} = \{(a, b) \mid a, b \in \mathcal{U}, b \in \mathcal{N}_a^k\}$. The edges in $\mathcal{G}_\mathcal{K}$ are undirected, and there are

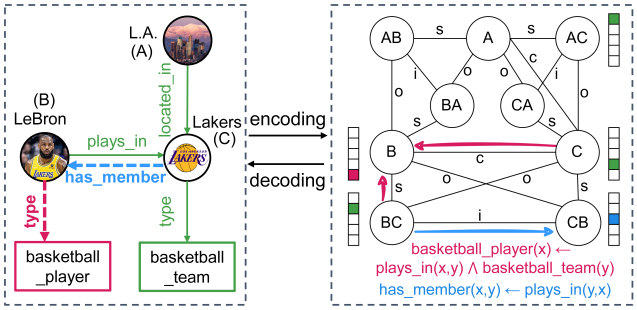

Figure 2: An example of an inference graph with the original KG. The left is the KG including three known facts (green) and two facts to predict (red and blue). The right is the inference graph, where A, B, and C stand for `L.A.`, `LeBron`, and `Lakers`, respectively.

five kinds of edges, self-loops, s (for "subject"), o (for "object"), i (for "inverse"), and c (for "correlated"). Each node has a self-loop. For each pair node (a, b) in $\mathcal{G}_\mathcal{K}$, there is a c-edge between a and b, an s-edge between a and (a, b), an o-edge between b and (a, b), and an i-edge between (a, b) and (b, a). An example inference graph of a KG with three facts $\mathsf{plays\_in}(\mathtt{LeBron}, \mathtt{Lakers})$, $\mathsf{locates\_in}(\mathtt{Lakers}, \mathtt{L.A.})$ and $\mathsf{basketball\_team}(\mathtt{Lakers})$ is shown in Figure 2.

Note that our definition of the inference graph is different from that in (Cucala et al. 2021) which only encodes pairs of entities that are directly connected in $\mathcal{K}$, we encode pairs of entities that are indirectly connected via k-hops with k being a hyper-parameter. For example, $(\mathtt{LeBron}, \mathtt{L.A.})$ is a pair node in the above example, which may be omitted in the graph of (Cucala et al. 2021).

We assume the binary and unary predicates in $\mathcal{K}$ have a fixed order with the binary ones listed before the unary ones, that is, $p_1, \ldots, p_\eta, c_{\eta+1}, \ldots, c_d$, where η is the number of binary predicates and $d - \eta$ is the number of unary predicates. Each node in $\mathcal{G}_\mathcal{K}$ has a feature vector in $\mathbb{R}^d$, and each (binary or unary) predicate has a unique index in the feature vector. For a vector $\mathbf{h}$, $\mathbf{h}_i$ is the i-th element of $\mathbf{h}$. For each node v in $\mathcal{G}_\mathcal{K}$, its feature vector is initialised, denoted $\mathbf{h}_v^{(0)}$, as follows: for a pair node (a, b), $(\mathbf{h}_{a,b}^{(0)})_i = 1$ if $p_i(a, b) \in \mathcal{K}$, and $(\mathbf{h}_{a,b}^{(0)})_i = 0$ otherwise. Similarly, for a singleton node a, $(\mathbf{h}_a^{(0)})_j = 1$ if $c_j(a) \in \mathcal{K}$, and $(\mathbf{h}_a^{(0)})_j = 0$ otherwise. Intuitively, the feature vector of a singleton node represents facts about the types of the entities, while that of a pair node represents facts about the relations between these entities. The values of $(\mathbf{h}_{a,b}^{(0)})_i$ for $i > \eta$ and $(\mathbf{h}_a^{(0)})_j$ for $j \leq \eta$ do not have meaning and they will be set to 0.

The feature vectors in $\mathcal{G}_\mathcal{K}$ will be processed by a GNN model $\mathcal{M}$ of $L \geq 1$ layers to perform reasoning, and the results can be easily decoded. We set a hyper-parameter $0 \leq \delta \leq 1$ as the threshold. Let $\mathcal{M}(\mathcal{K})$ be a set of facts consisting the fact $p_i(a, b)$ for each $1 \leq i \leq \eta$ and each pair node (a, b) such that $(\mathbf{h}_{a,b}^{(L)})_i \geq \delta$, and the fact $c_j(a)$ for each $\eta < j \leq d$ and each single node a such that $(\mathbf{h}_a^{(L)})_j \geq \delta$. Intuitively, $\mathcal{M}(\mathcal{K})$ consists of the results of reasoning.

Our Monotonic GNN

Similar to R-GCN (Schlichtkrull et al. 2018), in each layer of our GNN, the feature vector of each node is updated by aggregating messages from its neighbouring nodes with different weights applied for different edges. The weights for self-loops are stored in a learnable parameter matrix $\mathbf{A}^{(l)}$ for each layer $1 \leq l \leq L$, and those for the other four kinds of edges in learnable parameter matrices $\mathbf{B}_{\mathsf{s}}^{(l)}$, $\mathbf{B}_{\mathsf{o}}^{(l)}$, $\mathbf{B}_{\mathsf{i}}^{(l)}$, and $\mathbf{B}_{\mathsf{c}}^{(l)}$, respectively. Let $E = \{\mathsf{s}, \mathsf{o}, \mathsf{i}, \mathsf{c}\}$ denote the above four kinds of edges, and for a node v in $\mathcal{G}_{\mathcal{K}}$, let $\mathcal{N}_{v,\epsilon}$ be the set of neighbouring nodes in $\mathcal{G}_{\mathcal{K}}$ connected to v via edge of type $\epsilon \in E$. Let $\mathbb{R}^+$ denote the set of non-negative real numbers. The feature vector of node v in the l-th layer is given by

$$\mathbf{h}_v^{(l)} = \sigma\left(\mathbf{A}^{(l)}\mathbf{h}_v^{(l-1)} + \sum_{\epsilon \in E}\sum_{u \in \mathcal{N}_{v,\epsilon}} \mathbf{B}_{\epsilon}^{(l)}\mathbf{h}_u^{(l-1)} + \mathbf{b}^{(l)}\right) \quad (5)$$

where $\mathbf{A}^{(l)}$ and $\mathbf{B}_{\epsilon}^{(l)}$'s are over $\mathbb{R}^+$ of dimension $d \times d$, $\mathbf{b}^{(l)}$ is the bias vector over $\mathbb{R}^+$ with dimension d, and σ is the activation function $\mathsf{ClippedReLU}_{0,1}(x) = \min(\max(0, x), 1)$. Note that the size of our model $\mathcal{M}$ (i.e., the dimensions of the parameter matrices and vectors) only depends on the number of predicates, and thus $\mathcal{M}$ can be applied to any datasets with the same predicates.

For the self-loop to simulate the persistence of facts, the diagonal elements of $\mathbf{A}^{(l)}$ are set to 1. Also, for convenience in regulating the parameters (with rule confidence degrees), all the elements of $\mathbf{A}^{(l)}$ and $\mathbf{B}_{\epsilon}^{(l)}$'s are within the range of $[0, 1]$. We apply further masks to $\mathbf{A}^{(l)}$ and $\mathbf{B}_{\epsilon}^{(l)}$'s to ensure the updated feature vectors are still meaningful; that is, some values are forced to 0.

As the parameter matrices are non-negative and the activation function is monotonically increasing, our GNN model satisfies monotonicity as (Cucala et al. 2021), i.e., $\mathcal{M}(\mathcal{K}) \subseteq \mathcal{M}(\mathcal{K} \cup \mathcal{K}')$ for any KGs $\mathcal{K}$ and $\mathcal{K}'$. In addition, our GNN model satisfies $\mathcal{K} \subseteq \mathcal{M}(\mathcal{K})$, that is, facts are preserved during our GNN reasoning. This can be seen from the following argument: $\mathbf{A}^{(l)}\mathbf{h}_v^{(l-1)} \geq \mathbf{h}_v^{(l-1)}$ for each node v, as the diagonal elements of $\mathbf{A}^{(l)}$ are 1; and hence, $(\mathbf{h}_v^{(l-1)})_i \geq \delta$ implies $(\mathbf{h}_v^{(l)})_i \geq \delta$.

Rule Encoding and Decoding

We want to inject rules to guide the training of our GNN, as well as extract (possibly new) rules to explain the inference of the trained GNN. To this end, we explore connections between the parameters of our GNN and rules, and leverage such connections for both rule injection and extraction.

For example, consider a rule $p_i(x, y) \leftarrow p_j(x, y)$ in Figure 3. For each pair of entities $a, b \in \mathcal{E}$, if $p_j(a, b)$ is a fact in the KG, we want the GNN model to infer $p_i(a, b)$. From our KG encoding and decoding, $p_j(a, b)$ being a fact means the feature vector of a is initialised in a way that $(\mathbf{h}_{a,b}^{(0)})_j = 1$, and $p_i(a, b)$ is inferred only if $(\mathbf{h}_{a,b}^{(L)})_i \geq \delta$. From Equation (5), $(\mathbf{h}_{a,b}^{(1)})_i \geq \sigma(\mathbf{A}^{(1)}\mathbf{h}_{a,b}^{(0)})_i$ and $(\mathbf{A}^{(1)}\mathbf{h}_{a,b}^{(0)})_i \geq \mathbf{A}_{i,j}^{(1)} * (\mathbf{h}_{a,b}^{(0)})_j$. When $(\mathbf{h}_{a,b}^{(0)})_j = 1$, that is $(\mathbf{h}_{a,b}^{(1)})_i \geq \sigma(\mathbf{A}_{i,j}^{(1)})$. Hence, we can regularise the value of $\mathbf{A}_{i,j}^{(1)}$ to be sufficiently large to make $(\mathbf{h}_{a,b}^{(1)})_i$ greater than δ. Since $\mathbf{A}_{i,j}^{(1)}$ is independent of the dataset, it enforces $p_i(a, b)$ to hold whenever $p_j(a, b)$ does, for arbitrary $a, b \in \mathcal{E}$; that is, the rule $p_i(x, y) \leftarrow p_j(x, y)$ is encoded. Intuitively, such parameters in our GNN "simulate" the rule by the message passing along the self-loop edge in $\mathcal{G}_{\mathcal{K}}$.

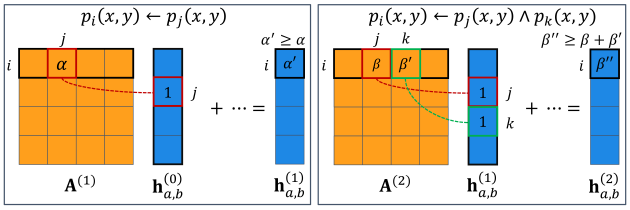

Figure 3: Examples rules and impact on GNN parameters.

ID	Rule	GNN Parameters
R1	$p_i(x, y) \leftarrow p_j(x, y)$	$\mathbf{A}_{i,j}^{(1)} \geq \xi_1$
R2	$p_i(x, y) \leftarrow p_j(y, x)$	$(\mathbf{B}_{\mathsf{i}}^{(1)})_{i,j} \geq \xi_1$
R3	$c_i(x) \leftarrow c_j(x)$	$\mathbf{A}_{i,j}^{(1)} \geq \xi_1$
R4	$c_i(x) \leftarrow p_j(x, y)$	$(\mathbf{B}_{\mathsf{s}}^{(1)})_{i,j} \geq \xi_1$
R5	$c_i(x) \leftarrow p_j(y, x)$	$(\mathbf{B}_{\mathsf{o}}^{(1)})_{i,j} \geq \xi_1$

Table 1: Connection between rules and GNN parameters.

The connection between the parameters of our GNN and five types of rules is shown in Table 1, where ξ_1 is the threshold value for the corresponding rule to be considered to hold.

For example in Figure 2, the reasoning of an R2 rule

$$\mathsf{has\_member}(x, y) \leftarrow \mathsf{plays\_in}(y, x)$$

can be simulated by the message passing via the i-edges in the inference graph. Hence, the rule can be encoded using the parameters in $\mathbf{B}_{\mathsf{i}}^{(1)}$. Rules R1 – R5 capture a relatively wide range of common rule patterns, including type and relation hierarchies of the forms $c_i(x) \leftarrow c_j(x)$ and $p_i(x, y) \leftarrow p_j(x, y)$, inverse and symmetric relations of the forms $p_i(x, y) \leftarrow p_j(y, x)$ and $p_i(x, y) \leftarrow p_i(y, x)$, and relation domain and range of the forms $c_i(x) \leftarrow p_j(x, y)$ and $c_i(x) \leftarrow p_j(y, x)$.

To inject rules of the forms R1 – R5 to guide the training of the GNN, we regularise the parameter matrices of its 1st layer. Instead of using a fixed universal threshold for all rules, we use the standard confidence degrees of the rules to regularise their corresponding parameters. That is, for instance, consider a rule r of the form R1, we regularise $\mathbf{A}_{i,j}^{(1)} \geq sc(r, \mathcal{K})$. Let $\mathbf{R}_k$ ($k = 1, \ldots, 5$) be a matrix of dimension $d \times d$ over $\mathbb{R}^+$, which stores the confidence degrees of all the rules r of the form R_k, i.e., $(\mathbf{R}_k)_{i,j} = sc(r, \mathcal{K})$ for i and j that are defined by the rule r in Table 1 and $(\mathbf{R}_k)_{i,j} = 0$ otherwise. For example, $\mathbf{R}_1$ corresponds to rules r is of the form R1: $p_i(x, y) \leftarrow p_j(x, y)$; thus, $(\mathbf{R}_1)_{i,j} = sc(r, \mathcal{K})$ for $1 \leq i, j \leq \eta$ and $(\mathbf{R}_1)_{i,j} = 0$ for $i > \eta$ or $j > \eta$. We require

ID	Rule	GNN Parameters
R6	$p_i(x,y) \leftarrow p_j(x,y) \wedge p_k(x,y)$	$\mathbf{A}^{(2)}_{i,j} + \mathbf{A}^{(2)}_{i,k} \geq \xi_2$, $\mathbf{A}^{(2)}_{i,j} < \xi_1$, and $\mathbf{A}^{(2)}_{i,k} < \xi_1$
R7	$c_i(x) \leftarrow c_j(x) \wedge c_k(x)$	$\mathbf{A}^{(2)}_{i,j} + \mathbf{A}^{(2)}_{i,k} \geq \xi_2$, $\mathbf{A}^{(2)}_{i,j} < \xi_1$, and $\mathbf{A}^{(2)}_{i,k} < \xi_1$
R8	$c_i(x) \leftarrow p_j(x,y) \wedge c_k(y)$	$(\mathbf{B}^{(2)}_{\mathsf{s}})_{i,j} + (\mathbf{B}^{(2)}_{\mathsf{c}})_{i,k} \geq \xi_2$, $(\mathbf{B}^{(2)}_{\mathsf{s}})_{i,j} < \xi_1$, and $(\mathbf{B}^{(2)}_{\mathsf{c}})_{i,k} < \xi_1$

Table 2: Connections between rules of length 2 and GNN parameters.

that $\mathbf{A}^{(1)}_{i,j} \geq (\mathbf{R}_1)_{i,j}$. Formally, the regularisation term is

$$\begin{aligned}\mathcal{L}_{reg} =& \| \max(\mathbf{R}_1 + \mathbf{R}_3 - \mathbf{A}^{(1)}, 0)\| + \| \max(\mathbf{R}_2 - \mathbf{B}^{(1)}_{\mathsf{i}}, 0)\| \\ &+ \| \max(\mathbf{R}_4 - \mathbf{B}^{(1)}_{\mathsf{s}}, 0)\| + \| \max(\mathbf{R}_5 - \mathbf{B}^{(1)}_{\mathsf{o}}, 0)\|. \quad (6)\end{aligned}$$

The regularisation term is to be combined with the loss about the facts in the KGs, denoted as $\mathcal{L}_{fact}$. We adopt the $\mathcal{L}_{fact}$ from (Liu et al. 2021), which uses a binary cross-entropy loss to enable the model to distinguish between correct and incorrect facts, as follows

$$\mathcal{L}_{fact} = -\sum_{i=1}^{n} [y_i \cdot \log(\hat{y_i}) + (1 - y_i) \cdot \log(1 - \hat{y_i})], \quad (7)$$

where y_i represents the true label of the i-th fact in the KG, with $y_i = 1$ indicating a positive example and $y_i = 0$ indicating a negative example. The variable $\hat{y_i}$ denotes the predicted probability that the i-th example is positive, as estimated by our GNN. The regularisation term is combined with $\mathcal{L}_{fact}$ by a hyper-parameter λ. The total loss for the training of our GNN with rule guidance is given by

$$\mathcal{L} = \mathcal{L}_{fact} + \lambda \mathcal{L}_{reg}. \quad (8)$$

After our GNN model $\mathcal{M}$ is trained, we can extract a set of rules $\mathcal{R}_{\mathcal{M}}$ from $\mathcal{M}$. Unlike most existing rule extraction approaches which enumerate and evaluate all candidate rules (Cucala et al. 2021; Liu et al. 2021), we "read off" rules directly from the parameters of $\mathcal{M}$, based on the connection we established in Table 1. In particular, a rule is considered to hold if the corresponding parameter is no less than the threshold ξ_1 which is a hyper-parameter.

Rules with Greater Lengths

Besides rules of the forms R1 – R5, more complex rules with lengths greater than 1 can be injected and extracted.

For example, consider a rule $r : p_i(x,y) \leftarrow p_j(x,y) \wedge p_k(x,y)$ in Figure 3. Similar to R1 rules, we want to enforce $p_i(a,b)$ to hold whenever $p_j(a,b)$ and $p_k(a,b)$ do, for arbitrary $a, b \in \mathcal{E}$. We achieve this by regularising the parameter matrices of the 2nd layer. More specifically, when $p_j(a,b)$ and $p_k(a,b)$ are both facts in the KG, we can regularise the value of $\mathbf{A}^{(2)}_{i,j} + \mathbf{A}^{(2)}_{i,k}$ to be sufficiently large to make $(\mathbf{h}^{(2)}_{a,b})_i \geq \delta$ and thus $p_i(a,b)$ to be inferred. Similarly, for rule extraction, the rule is considered to hold if $\mathbf{A}^{(2)}_{i,j} + \mathbf{A}^{(2)}_{i,k} \geq \xi_2$.

However, if the value of $\mathbf{A}^{(2)}_{i,j} + \mathbf{A}^{(2)}_{i,k}$ is large only because one of the addends is large, then the fact may be inferred from shorter rules $r' : p_i(x,y) \leftarrow p_j(x,y)$ or $r'' : p_i(x,y) \leftarrow p_k(x,y)$ instead of $r : p_i(x,y) \leftarrow p_j(x,y) \wedge p_k(x,y)$. And if r' or r'' hold, r is redundant, as it can be logically implied by either of them. To avoid this issue, we additionally require that $\mathbf{A}^{(2)}_{i,j} < \xi_1$ and $\mathbf{A}^{(2)}_{i,k} < \xi_1$.

We present the connection between the GNN parameters and three common rules of length 2 in Table 2. For example in Figure 2, the reasoning of an R8 rule

$$\mathsf{basketball\_player}(x) \leftarrow \mathsf{plays\_in}(x,y) \wedge \mathsf{basketball\_team}(y)$$

can be simulated by aggregating the messages passed via both of the s- and c-edges in the inference graph, whereas the value contributed from either the s- or the c-edge alone is insufficient to derive x is a basketball player. The rule can be encoded by combining the parameters in $\mathbf{B}^{(2)}_{\mathsf{s}}$ and $\mathbf{B}^{(2)}_{\mathsf{c}}$.

Rules R1 – R8 can express most of the description logic axioms in $\mathcal{ELHI}$ normal forms (Baader et al. 2017). They capture the rules extracted in (Liu et al. 2021) except for the relation chains (a.k.a. closed-path rules). We note that while our approach does not support encoding of relation chains, they can still be extracted from our GNN model via the rule extraction method in (Liu et al. 2021) by enumerating and evaluating all such rules.

Again, we use the standard confidence degrees of the rules to regularise the parameters for the GNN training. For instance, consider a rule r of the form R6, we require $\mathbf{A}^{(2)}_{i,j} + \mathbf{A}^{(2)}_{i,k} \geq sc(r, \mathcal{K})$, $\mathbf{A}^{(2)}_{i,j} < \xi_1$, and $\mathbf{A}^{(2)}_{i,k} < \xi_1$. Note that some of the constraints to the parameters may contradict each other, for example, when $sc(r, \mathcal{K}) > 2\xi_1$. In this case, we select rules to ensure a consistent upper and lower bound can be derived for each parameter.

Finally, we establish the soundness of the extracted rules. For our GNN model $\mathcal{M}$ satisfying $\delta \leq \xi_1$ and $\delta \leq \xi_2$, let $\mathcal{R}_{\mathcal{M}}$ be the rules of the forms R1 – R8 extracted from $\mathcal{M}$.

Proposition 1. *For each KG $\mathcal{K}$, $\mathcal{R}_{\mathcal{M}}(\mathcal{K}) \subseteq \mathcal{M}(\mathcal{K})$.*

Proof. We show for each fact $f \in \mathcal{R}_{\mathcal{M}}(\mathcal{K})$, $f \in \mathcal{M}(\mathcal{K})$. Suppose f is derived by a rule r with a substitution θ. We show the case of r being an R8 rule, and the cases when r is of the forms R1 – R7 can be shown in a similar way. Suppose $\theta(r)$ is $c_i(a) \leftarrow p_j(a,b) \wedge c_k(b)$. Then, $p_j(a,b)$ and $c_k(b)$ are both in $\mathcal{K}$. By the KG encoding, $(\mathbf{h}^{(0)}_{a,b})_j = 1$ and $(\mathbf{h}^{(0)}_{b})_k = 1$. By the monotonicity of the our GNN, $(\mathbf{h}^{(1)}_{a,b})_j \geq 1$ and $(\mathbf{h}^{(1)}_{b})_k \geq 1$. From Equation (5), $(\mathbf{h}^{(2)}_{a})_i \geq \sigma(\mathbf{B}^{(2)}_{\mathsf{s}}\mathbf{h}^{(1)}_{a,b} + \mathbf{B}^{(2)}_{\mathsf{c}}\mathbf{h}^{(1)}_{b})$ and $\mathbf{B}^{(2)}_{\mathsf{s}}\mathbf{h}^{(1)}_{a,b} + \mathbf{B}^{(2)}_{\mathsf{c}}\mathbf{h}^{(1)}_{b} \geq (\mathbf{B}^{(2)}_{\mathsf{s}})_{i,j} * (\mathbf{h}^{(1)}_{a,b})_j + (\mathbf{B}^{(2)}_{\mathsf{c}})_{i,k} * (\mathbf{h}^{(1)}_{b})_k$. That is, $(\mathbf{h}^{(1)}_{a})_i \geq \sigma((\mathbf{B}^{(2)}_{\mathsf{s}})_{i,j} + (\mathbf{B}^{(2)}_{\mathsf{c}})_{i,k})$. From Table 2, $r \in \mathcal{R}_{\mathcal{M}}$ only if $(\mathbf{B}^{(2)}_{\mathsf{s}})_{i,j} + (\mathbf{B}^{(2)}_{\mathsf{c}})_{i,k} \geq \xi_2$; that is, $(\mathbf{h}^{(1)}_{a})_i \geq \xi_2 \geq \delta$. By the monotonicity of the GNN, $f \in \mathcal{M}(\mathcal{K})$. □

Metrics	Methods	FB15K-237 v1	FB15K-237 v2	FB15K-237 v3	FB15K-237 v4	NELL-995 v1	NELL-995 v2	NELL-995 v3	NELL-995 v4	INDIGO-BM
Precision	DRUM	58.2	62.8	63.6	66.4	98.1	55.2	68.5	57.6	58.2
	NCRL	**99.7**	99.1	**99.3**	**99.5**	43.5	99.5	98.1	99.3	**99.6**
	MGNN	99.4	**99.4**	99.0	**99.5**	97.9	**99.7**	**99.9**	**99.7**	99.4
	INDIGO	94.7	96.1	96.1	95.2	76.9	91.8	96.2	75.6	98.1
	Ours	97.3	96.6	98.7	98.3	**98.9**	98.4	97.1	92.8	98.5
Recall	DRUM	54.6	63.2	62.4	55.2	57.8	42.2	45.9	29.4	54.6
	NCRL	28.8	35.1	31.6	33.4	73.0	57.1	57.1	**68.0**	16.9
	MGNN	32.7	41.2	34.6	35.2	80.0	48.3	55.3	57.2	33.0
	INDIGO	77.3	85.9	87.8	89.4	92.0	72.8	87.5	33.4	90.5
	Ours	**83.9**	**92.3**	**91.0**	**90.9**	**100.0**	**73.1**	**90.5**	63.2	**97.4**
Accuracy	DURM	57.5	62.6	63.0	63.3	78.4	53.5	62.1	53.3	57.5
	NCRL	64.3	67.4	65.7	66.6	39.2	78.4	78.0	**83.7**	58.4
	MGNN	66.2	70.5	67.1	67.5	89.1	74.1	77.6	78.5	66.4
	INDIGO	86.5	91.2	92.1	92.4	82.1	83.1	92.0	61.3	94.4
	Ours	**90.8**	**94.5**	**94.9**	**94.7**	**99.4**	**86.0**	**93.9**	79.1	**98.0**
F1 Score	DRUM	56.3	62.9	62.9	60.2	72.7	47.7	54.9	38.7	56.3
	NCRL	44.7	51.9	47.9	50.0	54.5	72.6	72.2	**80.7**	28.9
	MGNN	49.2	58.3	51.2	52.0	88.1	65.1	71.1	72.7	49.6
	INDIGO	85.1	90.8	91.8	92.2	83.7	81.2	91.6	46.4	94.1
	Ours	**90.1**	**94.4**	**94.7**	**94.5**	**99.5**	**83.9**	**93.7**	75.2	**98.0**
AUC	DRUM	66.3	70.8	67.5	69.2	90.2	61.2	68.7	61.8	66.3
	NCRL	64.3	67.3	65.6	66.6	41.7	78.5	77.4	83.8	58.4
	MGNN	66.3	70.6	67.2	67.6	89.5	74.1	77.6	78.6	66.5
	INDIGO	93.8	96.7	97.5	97.3	80.2	90.8	96.6	72.2	99.0
	Ours	**96.8**	**97.3**	**98.8**	**97.9**	**99.6**	**97.1**	**98.1**	**92.1**	**99.6**

Table 3: Inductive link prediction results (in percentage) on 9 benchmarks. Best results are in **bold**.

Experiments

We have evaluated our approach on inductive link prediction tasks. We have also analysed the number and the quality of extracted rules. The experiments are designed to validate the following statements:

- Our GNN exhibits better performance compared to explainable methods in inductive link prediction over common benchmarks (ref. Table 3).
- Our rule-guidance approach is effective, evidenced by the quality (measured by standard confidence) of the extracted rules from our GNN (ref. Figure 4).
- Our rule encoding method is effective, shown by the high correlation between the encoded and decoded rules (ref. Figure 5).

All experiments were performed on an Intel(R) Xeon(R) machine with 2 NVIDIA GeForce RTX 4090 GPUs. The implementation of our approach can be found at https://github.com/bohemianc/rule-guided-gnns.

Benchmarks and Baselines

We adopt commonly used benchmarks for inductive link prediction for our evaluation (Teru, Denis, and Hamilton 2020; Liu et al. 2021), including 8 datasets constructed from FB15K-237 (Bordes et al. 2013) and NELL-995 (Xiong, Hoang, and Wang 2017). There are 4 datasets (called versions) for each of these benchmarks, constructed by completely separating the entities in the test set from those in the training set. As our model can encode and process type information, we used the datasets with triples about entity types, which are proposed in (Ma et al. 2023). We also include the benchmark INDIGO-BM (Liu et al. 2021), which contains triples about types.

We have included two groups of state-of-the-art explainable methods as baselines to evaluate the performance of our approach. The first group comprises end-to-end rule learners, including DRUM (Sadeghian et al. 2019) and NCRL (Cheng, Ahmed, and Sun 2022). The second group consists of GNN models for inductive link prediction that support rule extraction to explain the predictions, including MGNN (Cucala et al. 2021) and INDIGO (Liu et al. 2021).

Inductive Link Prediction

We evaluate the performance of our model in inductive link prediction using standard classification metrics, including precision, recall, accuracy, F1 Score, and *the area under the precision-recall curve* (AUC). We evaluate each system on a given benchmark over 10 runs with independently sampled sets of negative examples, and report the average.

The results of inductive link prediction in Table 3 show that our method attains the highest scores on most benchmarks in recall, accuracy, F1 score, and AUC, while remaining competitive in precision. While NCRL and MGNN exceed 99% in precision, our method achieves over 97% in most cases. Notably, our method significantly outperforms DRUM, NCRL, and MGNN in recall and slightly surpasses

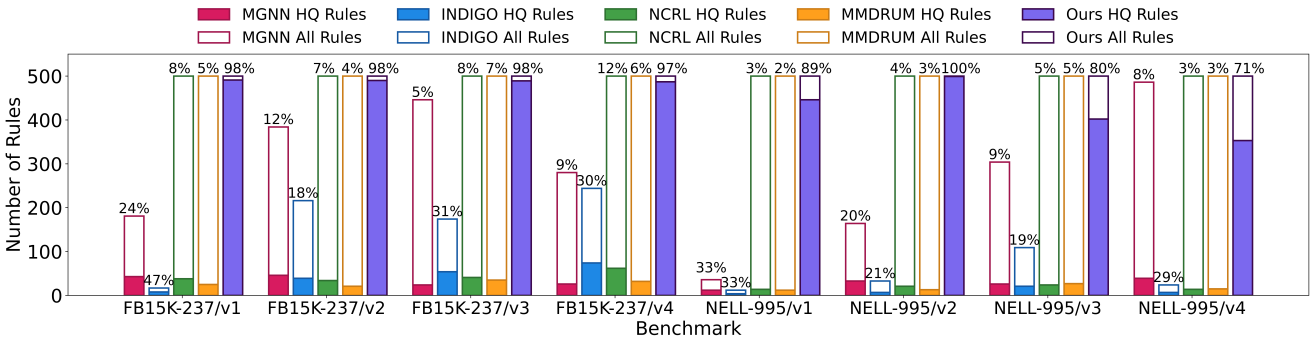

Figure 4: Percentage of high-quality rules (HQ rules) in all extracted rules for different methods on 8 benchmarks.

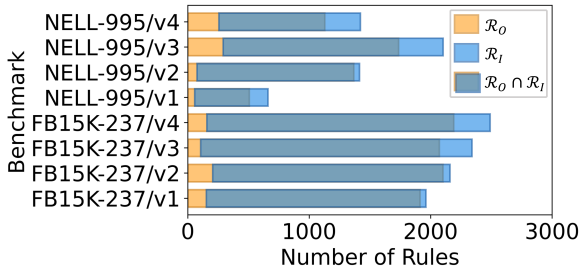

Figure 5: Correlations between the input ($\mathcal{R}_I$) and output ($\mathcal{R}_O$) high-quality rules of our model.

INDIGO. It also secures the highest accuracy and F1 scores, demonstrating its overall superiority. Moreover, our method consistently achieves the top AUC values across all benchmarks, with most scores reaching 97%. This analysis underscores the effectiveness of our method in inductive link prediction, particularly as an explainable approach, delivering both high performance and explainability.

We have analysed the effect of (different design choices of) rule guidance by comparing their F1 scores on FB15K-237. Here, applying all the rules together means at each layer both Tables 1 and 2 are used in the regulation term.

Methods	v1	v2	v3	v4
without rule guidance	88.6	93.7	93.8	94.5
apply all the rules together	83.4	90.9	93.1	93.0
our current model	**90.1**	**94.4**	**94.7**	**94.5**

Table 4: F1 scores on FB15K-237.

The results clearly show the benefit of rule guidance, yet applying all the rules together may confuse the model.

Quality of Extracted Rules

We evaluated the quality of extracted rules from our method and the baselines with the quality of rules measured by their standard confidence degrees. As the published code for DRUM does not include the rule extraction component, we used MMDRUM (Wang et al. 2023) instead, a faithful version of DRUM. The results on the 8 datasets are illustrated in Figure 4. We report the number of rules extracted by each model, and when over 500 rules can be extracted from a model, we use the top 500 rules with the highest confidence degrees. We also report the percentage of high-quality (HQ) rules among the (up to 500) extracted rules, where a rule is considered high-quality if its confidence degree is not less than 0.4. We can see that our method consistently produces a higher percentage of high-quality rules across different benchmarks, suggesting a more effective rule-guided approach. Our method attains a percentage of above 90% in most cases and extracts over 500 rules on each benchmark. Although NCRL and MMDRUM also extracted many rules, the proportion of high-quality rules is relatively low.

Next, we compared the high-quality rules input into ($\mathcal{R}_I$) and output from ($\mathcal{R}_O$) our model. The numbers of rules in $\mathcal{R}_I$ and $\mathcal{R}_O$ and their overlaps are summarised in Figure 5. The significant overlaps indicate that most of the injected rules can be extracted after the training, and thus the inference of the model is effectively guided by the injected rules. Additionally, the model discovered high-quality rules beyond the injected ones.

Conclusion

In this paper, we have proposed a new family of GNNs whose training can be guided by rules. This is achieved through a connection between the parameters of our GNN with various forms of rules. Specifically, our approach allows (i) effective rule injection to guide the GNN training, (ii) efficient rule extraction by directly "reading off" the GNN parameters, (iii) a comparison between the two sets of rules to analyse the effectiveness of our rule guidance, and (iv) the extracted rules that are both sound and plausible. Experiments show that our model outperforms existing explainable models in inductive link prediction on common benchmarks, and the quality of the extracted rules is outstanding. Interesting future work includes extending our approach for more complex rules and neural networks.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under grant 61976153.

References

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Chen, S.; Cai, Y.; Fang, H.; Huang, X.; and Sun, M. 2024. Differentiable neuro-symbolic reasoning on large-scale knowledge graphs. *Advances in Neural Information Processing Systems*, 36.

Cheng, K.; Ahmed, N.; and Sun, Y. 2022. Neural Compositional Rule Learning for Knowledge Graph Reasoning. In *International Conference on Learning Representations*.

Cucala, D. J. T.; Grau, B. C.; Kostylev, E. V.; and Motik, B. 2021. Explainable GNN-Based Models over Knowledge Graphs. In *International Conference on Learning Representations*.

Cucala, D. J. T.; Grau, B. C.; and Motik, B. 2022. Faithful Approaches to Rule Learning. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, 484–493.

Cucala, D. T.; Grau, B. C.; Motik, B.; and Kostylev, E. V. 2023. On the Correspondence Between Monotonic Max-Sum GNNs and Datalog. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, 658–667.

Galárraga, L. A.; Teflioudi, C.; Hose, K.; and Suchanek, F. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, 413–422.

Guo, S.; Wang, Q.; Wang, L.; Wang, B.; and Guo, L. 2016. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, 192–202.

Guo, S.; Wang, Q.; Wang, L.; Wang, B.; and Guo, L. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *AAAI*, volume 32.

Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G. D.; Gutierrez, C.; Kirrane, S.; Gayo, J. E. L.; Navigli, R.; Neumaier, S.; et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4): 1–37.

Liu, S.; Grau, B.; Horrocks, I.; and Kostylev, E. 2021. Indigo: Gnn-based inductive knowledge graph completion using pair-wise encoding. *Advances in Neural Information Processing Systems*, 34: 2034–2045.

Liu, T.; Lv, Q.; Wang, J.; Yang, S.; and Chen, H. 2024. Learning Rule-Induced Subgraph Representations for Inductive Relation Prediction. *Advances in Neural Information Processing Systems*, 36.

Ma, S.; Wang, Z.; Wang, K.; and Zhuang, Z. 2023. Type-enhanced Inductive Knowledge Graph Completion. In *ISWC (Posters/Demos/Industry)*.

Nayyeri, M.; Xu, C.; Alam, M. M.; Lehmann, J.; and Yazdi, H. S. 2021. LogicENN: a neural based knowledge graphs embedding model with logical rules. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6): 7050–7062.

Niu, G.; Zhang, Y.; Li, B.; Cui, P.; Liu, S.; Li, J.; and Zhang, X. 2020. Rule-guided compositional representation learning on knowledge graphs. In *AAAI*, volume 34, 2950–2958.

Omran, P. G.; Wang, K.; and Wang, Z. 2019. An embedding-based approach to rule learning in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 33(4): 1348–1359.

Pan, Y.; Liu, J.; Zhao, T.; Zhang, L.; Lin, Y.; and Dong, J. S. 2024. A Symbolic Rule Integration Framework with Logic Transformer for Inductive Relation Prediction. In *Proceedings of the ACM on Web Conference 2024*, 2181–2192.

Qu, M.; Chen, J.; Xhonneux, L.-P.; Bengio, Y.; and Tang, J. 2020. RNNLogic: Learning Logic Rules for Reasoning on Knowledge Graphs. In *International Conference on Learning Representations*.

Sadeghian, A.; Armandpour, M.; Ding, P.; and Wang, D. Z. 2019. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Trans. Neural Networks*, 20(1): 61–80.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *ESWC*, 593–607.

Teru, K.; Denis, E.; and Hamilton, W. 2020. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, 9448–9457. PMLR.

Wang, X.; Cucala, D. J. T.; Grau, B. C.; and Horrocks, I. 2023. Faithful Rule Extraction for Differentiable Rule Learning Models. In *The Twelfth International Conference on Learning Representations*.

Wu, H.; Wang, Z.; Wang, K.; Omran, P. G.; and Li, J. 2023. Rule Learning over Knowledge Graphs: A Review. *TGDK*, 1(1): 7:1–7:23.

Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 564–573.

Yan, Z.; Ma, T.; Gao, L.; Tang, Z.; and Chen, C. 2022. Cycle representation learning for inductive relation prediction. In *International Conference on Machine Learning*, 24895–24910. PMLR.

Yang, F.; Yang, Z.; and Cohen, W. W. 2017. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30.

Zhang, S.; Zhang, J.; Song, X.; Adeshina, S.; Zheng, D.; Faloutsos, C.; and Sun, Y. 2023. Page-link: Path-based graph neural network explanation for heterogeneous link prediction. In *Proceedings of the ACM Web Conference 2023*, 3784–3793.