Path Spuriousness-aware Reinforcement Learning for Multi-Hop Knowledge Graph Reasoning

Chunyang Jiang[1,2], Tianchen Zhu[1,2,4], Haoyi Zhou[1,3], Chang Liu[3], Ting Deng[1,2], Chunming Hu[1,3] and Jianxin Li[1,2]✉
[1]SKLSDE, Beihang University
[2]School of Computer Science and Engineering, Beihang University
[3]School of Software, Beihang University
[4]Shenyuan Honors College, Beihang University
{jiangcy,zhutc,zhouhy,liuc,dengting,hucm,lijx}@act.buaa.edu.cn

Abstract

Multi-hop reasoning, a prevalent approach for query answering, aims at inferring new facts along reasonable paths over a knowledge graph. Reinforcement learning (RL) methods can be adopted by formulating the problem into a Markov decision process. However, common suffering within RL-based reasoning models is that the agent can be biased to spurious paths which coincidentally lead to the correct answer with poor explanation. In this work, we take a deep dive into this phenomenon and define a metric named Path Spuriousness (PS), to quantitatively estimate to what extent a path is spurious. Guided by the definition of PS, we design a model with a new reward that considers both answer accuracy and path reasonableness. We test our method on five datasets and experiments reveal that our method considerably enhances the agent's capacity to prevent spurious paths while keeping comparable to state-of-the-art performance.

1 Introduction

Knowledge Graph (KG), a set of structured facts about real-world human knowledge, is utilized in numerous downstream NLP applications (Hildebrandt et al., 2020; Zhang and Yao, 2022; Xu et al., 2021; Ma et al., 2021). Common suffering affecting many downstream tasks is KG incompleteness. A variable amount of facts are missing in practical KGs. KG reasoning, the process to derive new knowledge from KG (Ji et al., 2022; Zhang et al., 2022; Huang et al., 2022), is the way to address KG completion problem. A prevailing approach for KG reasoning is incorporating KG embedding (KGE), which maps entities and relations into a vector space (Bordes et al., 2013). Embedding-based models have great power in expressing semantic similarity of entities and relations, but usually lack explainability due to the high-dimension representation (Chen et al., 2020; Heo et al., 2022).

✉ The corresponding author is Jianxin Li.

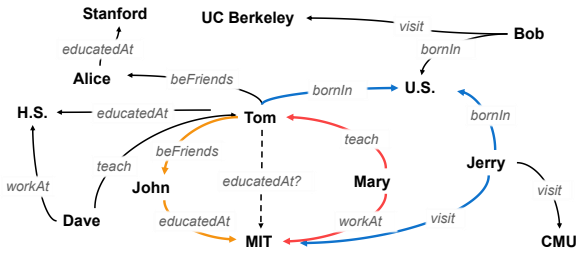

Figure 1: An example KG (we use *H.S.* to indicate a certain high school for short). Supposing the query is <*Tom, educatedAt, ?*> and *MIT* is a correct answer, three paths leading to *MIT* are of different PS.

An alternative approach is multi-hop reasoning, also referred to as path-based models (Lin et al., 2018), which infers new facts along existing paths in KG. Multi-hop reasoning offers explanations for its predictions by taking advantage of reasoning paths. Recently, reinforcement learning (RL) has been applied to multi-hop reasoning (Xiong et al., 2017). RL-based methods train an agent to walk over the KG and search for a path leading to the answer (Das et al., 2018). They have drawn surging attention in the past few years for their good prediction accuracy and excellent explainability.

However, most prevalent RL-based models are suffering from the spurious path problem (Guu et al., 2017). A spurious path reaches a correct answer merely by coincidence and has no logical relevance with its prediction, such as the blue path in Figure 1: we shouldn't say *Tom* was educated at *MIT* just because there exists one person (*i.e.*, *Jerry*) who was born in the same country (*i.e.*, *U.S.*) with *Tom* and happened to visit *MIT* before.

Multi-hop reasoning is a typical sparse reward scenario, where all actions except the final one will get no feedback during the decision process. That is to say, after the agent reached the correct answer following a spurious path, all actions along the decision trajectory will get positive rewards even if they are totally irrelevant to the query. As a

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3181–3192
May 2-6, 2023

consequence, the agent will be biased toward those wrong actions (Guu et al., 2017).

The spurious path problem causes severe damage to the explainability of RL-based models when paths found by the agent are needed to serve as evidence to explain the answer. Moreover, spurious paths may mislead the agent to learn a wrong policy and further harm the generalization ability of the model. Although some studies have noticed the spurious path problem and provided instinctive solutions, such as action drop (Lin et al., 2018) and rule guider (Lei et al., 2020; Hou et al., 2021), a quantitative estimation is still absent.

To address the above-mentioned problem, we define a new metric called *Path Spuriousness* (PS) to measure to what extent a path is spurious. The inspiration is that a spurious path is not a fake path, but an adventitious path offering the accurate answer for a certain query. That is to say, if we randomly exchange intermediate entities of the path and keep the relation order, it is unlikely to get the correct prediction for the same query. With PS, we can reflect the reasonableness of predictions made by multi-hop reasoning models. Specifically, answers obtained by following paths with low PS are much more reasonable and explainable than those by following paths with high PS.

With the definition of PS, we put forward a path spuriousness-aware reward for RL-based multi-hop reasoning models. Combined with correctness-guided rewards, our new reward leads the agent to not only obtain effective answers but also offer high-quality reasoning paths.

Our major contributions are concluded as follows: (1) We first propose a quantitative metric PS to measure the spuriousness of reasoning paths. (2) We design a new sophisticated reward shaping method by incorporating correctness-guided reward and PS-guided reward, which leads the agent to find effective answers following reasonable paths. (3) We first offer an empirical evaluation of the path spuriousness of multi-hop reasoning methods. Experiments show that our approach has a great improvement in avoiding spurious paths while keeping the prediction accuracy.

2 Related Work

In this section, we give outlines of two main related areas and discuss their connection to our method.

2.1 Knowledge Graph Embedding

KG embedding translates semantic features of entities and relations to vector space, to give answers directly by operations over query vectors. TransE (Bordes et al., 2013) did seminal work in leveraging KG embedding to solve the QA problem and becomes the base model for a series of algorithms (Ji et al., 2015; Wang et al., 2014; Lin et al., 2015). DisMult (Yang et al., 2015) proposes a unified learning framework for embedding models and introduces an approach to mine logic rules with learned relation embeddings. ComplEX (Trouillon et al., 2016) uses complex vectors to represent entities and relations, to handle asymmetric relations. ConvE (Dettmers et al., 2018) takes advantage of a convolutional neural network for knowledge embedding in a large graph. Despite the powerful representational ability shown by embedding-based models, they are limited in many scenarios for the lack of explainability (Roscher et al., 2020; Liu et al., 2017), as one-hop reasoning methods.

A promising approach is incorporating KG embedding as a reward shaping function for multi-hop reasoning methods, which is first adopted by MultiHop (Lin et al., 2018) and followed by us.

2.2 Multi-Hop Reasoning

Compared to embedding-based models, multi-hop reasoning models predict by inferring a path step by step. The property is exactly desired in scenarios where not only an answer is required, but also evidence is demanded to explain the answer. Reinforcement learning algorithms can be naturally deployed to multi-hop reasoning. DeepPath (Xiong et al., 2017) first takes REINFORCE as the generator of evidence paths which are fed to PRA (Brin, 1998) subsequently. MINERVA (Das et al., 2018) takes the lead to design an end-to-end RL-based multi-hop reasoning model to address query answering, whereas its accuracy of answering still falls behind state-of-the-art embedding models. MultiHop (Lin et al., 2018) incorporates pre-trained embedding-based models as a soft reward function to compensate for the incompleteness of KG and gets comparable performance to embedding-based models in several datasets.

Since the agent lacks logical insights into adopted paths, spurious paths are inevitable during training. Once a spurious path leading to the correct answer coincidentally is explored first, the agent will increasingly tend to choose actions fol-

lowing a spurious way (Guu et al., 2017). To address this problem, MultiHop (Lin et al., 2018) uses action drop to force the agent to explore diverse paths, hoping to mitigate the negative effect of spurious paths. RARL (Hou et al., 2021) exploits high-quality rules mined from datasets to supervise the policy of the agent. However, there is no quantitative measure for the effect of the above methods, due to the absence of the metric that scales path spuriousness and consequently no reward that can directly guide the agent to avoid spurious paths.

3 Preliminaries

In this section, we first give crucial definitions of several concepts. Then the reinforcement learning formula and reward shaping skills will be introduced, which is the foundation of our model.

3.1 Multi-hop KG Reasoning

Knowledge Graph. A knowledge graph (KG) is a directed graph $G = (\mathcal{E}, \mathcal{R}, \Psi)$ (Ji et al., 2022), where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations and Ψ is the mapping from relations to pairs of entities. A fact in knowledge graph G is an ordered triple $\delta =< e_s, r, e_o >$ satisfying that $e_s, e_o \in \mathcal{E}$, $r \in \mathcal{R}$, and $\Psi(r) =< e_s, e_o >$.

In this paper, we treat each edge in a KG as bidirectional and augment the KG with reversed edge (e_o, r^{-1}, e_s) if (e_s, r, e_o) is a fact in G. We call r^{-1} the inverse relation of r.

Multi-Hop Reasoning. Given a query $q =< e_s, r_q, ? >$ ($<?, r_q, e_o >$ is the same) and a KG G, multi-hop reasoning is to predict the absent object by finding an n-hop path $\tau =< e_s, r_1, e_2, r_2, \cdots, e_n, r_n, e_T >$ in G (Wan and Du, 2021), where the last entity e_T is the predicted answer. The path can also be represented as a predicate $\tau \equiv r_1(e_s, e_2) \wedge r_2(e_2, e_3) \wedge \cdots \wedge r_n(e_n, e_T)$.

Path Clause. For convenience, we use the path clause $H(\tau, q) \equiv \tau \rightarrow r_q(e_s, e_T) \equiv r_1(e_s, e_2) \wedge r_2(e_2, e_3) \wedge \cdots \wedge r_n(e_n, e_T) \rightarrow r_q(e_s, e_T)$ to indicate a reasoning path τ and its prediction, where τ is the clause body and $r_q(e_s, e_T)$ is the head.

In this paper, we assume that all facts in KG are correct. Then path clause $H(\tau, q)$ is true in KG G if τ is a path in G and $r_q(e_s, e_T)$ is a fact in G. We call $H(\tau, q)$ is *valid* in G if τ is a path in G.

Path Substitution. Path substitution of $H(\tau, q)$ is defined as $H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T} \equiv r_1(e_s, x_2) \wedge r_2(x_2, x_3) \wedge \cdots \wedge r_n(x_n, x_T) \rightarrow r_q(e_s, x_T)$, which is a path clause derived by replacing each e_i in clause $H(\tau, q)$ with x_i except e_s, *i.e.,* keeping e_s fixed.

We denote the body of $H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}$ as $B_{H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}}$. We call $H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}$ is a *valid* path substitution of $H(\tau, q)$ in KG G if the body $B_{H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}}$ is a path in G.

Note that there may be overlaps of entities of a path substitution with the original path clause. That is, a path substitution may not replace all entities in a path clause with new entities. In particular, each path clause $H(\tau, q)$ is a path substitution of itself.

Consider the KG in Figure 1. For query *<Tom, educatedAt, ?>*, the path in color red indicates a prediction *<Tom, educatedAt, MIT>*, and the path clause H=*teach*$^{-1}$*(Tom, Mary)* $\wedge$ *workAt(Mary, MIT)* $\rightarrow$ *educatedAt(Tom, MIT)*. Clearly, $H^{Mary,MIT}_{Dave,H.S.}$, i.e., *teach*$^{-1}$*(Tom, Dave)* $\wedge$ *workAt(Dave, H.S.)* $\rightarrow$ *educatedAt(Tom, H.S.)* is a valid path subsituition of H in the KG.

3.2 RL Formula

The RL algorithms can be naturally deployed to multi-hop reasoning by formulating it into a Markov decision process (MDP) (Puterman, 1994). Following MINERVA (Das et al., 2018), we adopt REINFORCE (Williams, 1992) algorithm. Key components of the architecture are as follows.

States. A state encodes the status quo as well as the origin goal. In step t, state $s_t = (e_t, e_s, r_q)$ is a triple, where e_t is the current entity, e_s is the start entity and r_q is the query relation.

Actions. Action space $\mathcal{A}_t = \{(r, e') | (e_t, r, e') \in G\}$ in state s_t consists of all pairs of outgoing edges and corresponding entities. That is, the valid actions are neighborhoods of the current entity e_t.

Transition. After the agent makes its decision $a_{t+1} = (r, e')$, the state of environment migrates from $s_t = (e_t, e_s, r_q)$ to $s_{t+1} = (e', e_s, r_q)$, on condition $(e_t, r, e') \in G$.

Policy. Agent's policy maps states to actions, which is usually implemented by the deep neural network. To exploit history information, the long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997) is adopted. History information h_t in step t is calculated by LSTM, taking as input previous history h_{t-1} and the last action $a_t = (r_t, e_t)$, as follows:

$$\mathbf{h}_0 = \text{LSTM}(0, \mathbf{a}_0) \tag{1}$$

$$\mathbf{h}_t = \text{LSTM}(\mathbf{h}_{t-1}, \mathbf{a}_t), \quad t > 0. \tag{2}$$

Here $\mathbf{a}_t$ is the embedding of action a_t and $\mathbf{h}_t$ is the embedding of history h_t . Policy network π_θ gives possibility distribution over $\mathcal{A}_t$:

$$\pi_\theta(a_{t+1}|s_t) = \sigma(\mathbf{A}_t\mathbf{W_2}\mathrm{ReLU}(\mathbf{W_1}[\mathbf{h_t};\mathbf{e_t};\mathbf{r_q}])), \tag{3}$$

where $\mathbf{A}_t$ is a matrix stacking all action embeddings, σ is *sigmoid* function and *ReLU* is rectified linear unit (Nair and Hinton, 2010). The agent chooses an action subjecting to the distribution given by the policy network.

Reward. The default binary reward R_b indicates whether the agent arrives at the correct answer. With the advantage of indicator function $\mathbb{I}$, binary reward can be written as follow:

$$R_b(e_T) = \mathbb{I}((e_s, r_q, e_T) \in G). \tag{4}$$

Optimization. To get the optimal parameters, the policy network is trained by maximizing the objective function:

$$J(\theta) = \mathbb{E}_{(e_s,r_q,e_o)\in G_q}\mathbb{E}_{(a_1,a_2,\cdots,a_T)\sim\pi_\theta}[R_T]. \tag{5}$$

$$R_T = R(s_T|s_1 = (e_s, e_s, r_q)). \tag{6}$$

$J(\theta)$ is the expected reward for all queries following policy π_θ. G_q denotes all query facts. REINFORCE is deployed to solve this optimization problem by iteratively updating θ as follows:

$$\nabla_\theta J(\theta) \approx \sum_{t=1}^{T} R_T \nabla_\theta \log \pi_\theta(a_t|s_t) \tag{7}$$

$$\theta' = \theta + \nabla_\theta J(\theta). \tag{8}$$

3.3 Reward Shaping

Under binary reward R_b, the agent receives a positive reward only when it reaches a correct answer. That is to say, the agent probably can't get any effective guidance in the early exploration stage.

To offset the incompleteness of KG, Multi-hop (Lin et al., 2018) proposes a reward shaping function R_s that provides a soft reward F_s between 0 and 1 other than 0 as R_b does, when the prediction is not in KG during the training process. $F_s(e_s, r_q, e_T)$ is implemented by a pre-trained embedding-based model such as ConvE (Dettmers et al., 2018) and ComplEX (Trouillon et al., 2016) to estimate the probability that (e_s, r_q, e_T) is true. Then R_s is defined as follows:

$$R_s(e_T)=R_b(e_T)+(1-R_b(e_T))\cdot F_s(e_s, r_q, e_T). \tag{9}$$

That is, given the answer e_T, if $(e_s, r_q, e_T) \in G$, the agent will get 1 as the final reward, otherwise the agent will get $F_s(e_s, r_q, e_T)$ calculated by a pre-trained embedding-based model.

4 Methodology

In this section, we discuss the feature of spurious paths and give a quantitative definition of PS. Then we introduce two kinds of reward functions that take PS into account.

4.1 Path Spuriousness Metric

Think over the example in Figure 1. Assuming <*Tom, educatedAt, MIT*> is true and can be deduced by the following three paths H_1, H_2, and H_3 (in color red, blue, and yellow, respectively):

(1) H_1≡*teach*$^{-1}$*(Tom, Mary)* ∧ *workAt(Mary, MIT)* → *educatedAt(Tom, MIT)*;
(2) H_2≡*bornIn(Tom, U.S.)*∧*bornIn*$^{-1}$*(U.S., Jerry*∧*visit(Jerry, MIT)*→*educatedAt(Tom, MIT)*;
(3) H_3≡*beFriends(Tom, John)*∧*educatedAt(John, MIT)*→*educatedAt(Tom, MIT)*.

However, not all of them can serve as valid evidence. Clearly, H_1 is solid enough since *Tom* must be educated exactly where his teacher teaches. H_2 is extremely spurious because there are hundreds of millions of people born in the same place as *Tom* and they visit tremendous schools, few of which *Tom* can attend. Though H_3 is spurious in logic, considering that many of *Tom*'s friends made acquaintances with him when they were in the same school, H_3 is much more valid than H_2.

It is a remarkable fact that these three clauses are all *true* in the KG of Figure 1, despite their variance in path spuriousness. The truth value of a valid path clause always keeps consistent with its prediction, regardless of whether its body is a reasonable path or not. The key difference between spurious paths and reasonable paths is that they have quite a few valid substitutions in a given KG, but fail to infer correct predictions. That is, the path spuriousness of a path clause H is up to the proportion of such valid substitutions. The larger the proportion is, the more spurious H is.

Path Spuriousness. Given a KG G and a path clause $H \equiv r_1(e_s, e_2) \wedge r_2(e_2, e_3) \wedge \cdots \wedge r_n(e_n, e_T) \rightarrow r_q(e_s, e_T)$, the PS of H is:

$$\mathrm{PS}(H) = \mathbb{P}(v(H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}) = 0), \tag{10}$$

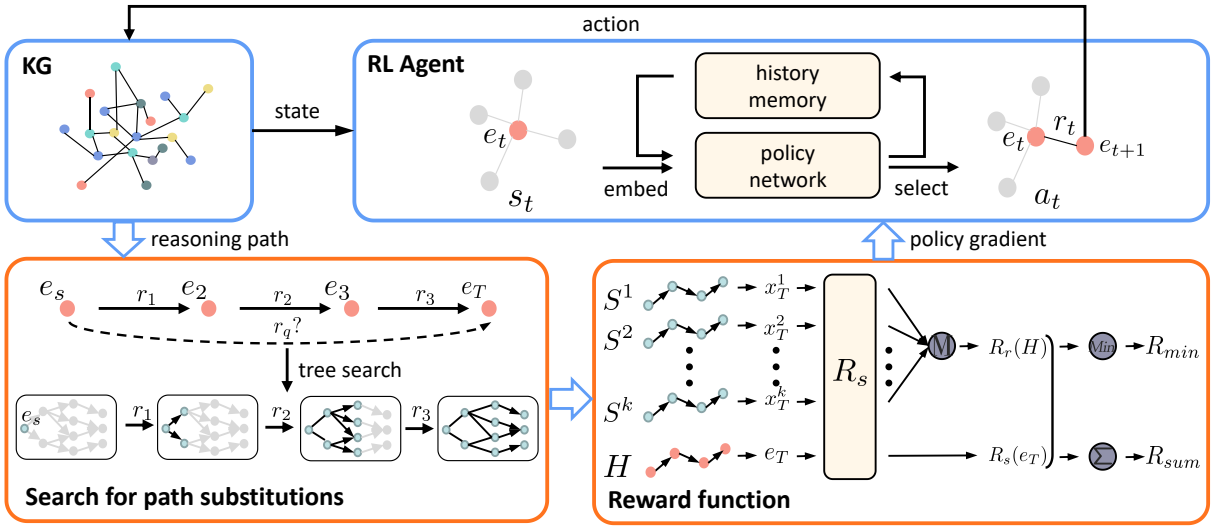

Figure 2: An overlook of the training process. Starting from entity e_s, the agent walks step by step to obtain a reasoning path H where e_T denotes the predicted answer. All substitutions of H are derived by searching over the KG. We use S^i to represent the i-th substitution. R_{min} and R_{sum} indicate our two reward shaping variants.

which is equal to:

$$\mathrm{PS}(H) = 1 - \mathbb{E}(v(H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T})) \tag{11}$$

where $H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}$ is a valid substitution of H, v is the value function mapping predicate space to $\{0,1\}$, and $X = (x_2,\cdots,x_n,x_T)$ are random entities sampled from the ideal entity set. In practice, the ideal entity set is usually unknown and hard to estimate, so we propose a computation viable metric based on frequency to approximate Equation 11:

$$\mathrm{PS}(H) \approx 1 - \frac{\sum_{x_2,\cdots,x_n,x_T \in \mathcal{E}} v(H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T})}{\sum_{x_2,\cdots,x_n,x_T \in \mathcal{E}} v(B_{H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}})} \tag{12}$$

where $B_{H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}}$ is the body part of $H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}$, and $\mathcal{E}$ is the entity set of G.

Take the KG in Figure 1 for example.

(1) H_1 (in color red) has only one valid substitution *teach*$^{-1}$*(Tom, Dave)* $\wedge$ *workAt(Dave, H.S.)* $\rightarrow$ *educatedAt(Tom, H.S.)* except H_1, which is true in the KG. Counting H_2 itself in, $\mathrm{PS}(H_2)$=1−2/2=0.

(2) H_2 (in color blue) has two valid substitutions except H_1: *bornIn(Tom, U.S.)* $\wedge$ *bornIn*$^{-1}$*(U.S.,Jerry)* $\wedge$ *visit(Jerry,CMU)* $\rightarrow$ *educatedAt(Tom, CMU)* and *bornIn(Tom, U.S.)* $\wedge$ *bornIn*$^{-1}$*(U.S.,Bob)* $\wedge$ *visit(Bob,UC Berkeley)* $\rightarrow$ *educatedAt(Tom, UC Berkeley)*, but both of them are false in the KG. Then we have $\mathrm{PS}(H_1)$=1−1/3=2/3.

(3) H_3 (in color yellow) has one valid substitution *beFriends(Tom, Alice)* $\wedge$ *educatedAt(Alice, Stanford)* $\rightarrow$ *educatedAt(Tom, Stanford)* except H_1, which is false in the KG. As a result, $\mathrm{PS}(H_3)$=1−1/2=1/2.

4.2 Path Spuriousness-Based Reward

Both R_b and R_s are devoted to the correctness of the terminal entity, ignoring the spuriousness of reasoning paths. To address this, we design two novel rewards combining the answer correctness as well as the path spuriousness.

First, a path score function $F_p(H)$ is required to score the spuriousness degree of H. The most straightforward way is taking the definition of PS as F_p, which is effective when KG is closed (Tanon et al., 2017). However, taking the incompleteness of KG into consideration, we incorporate the soft reward R_s with Equation 12 to build $F_p(H)$:

$$F_p(H) = 1 - R_r(H) \tag{13}$$

$$R_r(H) = \frac{\sum_{x_2,\cdots,x_n,x_T \in \mathcal{E}} v(B_{H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}}) R_s(x_T)}{\sum_{x_2,\cdots,x_n,x_T \in \mathcal{E}} v(B_{H^{e_2,\cdots,e_n,e_T}_{x_2,\cdots,x_n,x_T}})}. \tag{14}$$

For simplicity, we use $R_r(H) = 1 - F_p(H)$ to indicate the reasonableness of H.

Because of the complexity of semantic knowledge, even a reasonable path may lead to a wrong answer in some special cases. Therefore, a rational reward has to be a fusion of the answer accuracy and the path reasonableness. We demonstrate two

kinds of combinations between R_s and R_r, and discuss their performance in the experiment section.

One is taking the *minimum* between accuracy and reasonableness:

$$R_{min}(H, e_T) = min(R_s(e_T), R_r(H)). \quad (15)$$

The equation is based on intuitive thinking that, we introduce PS-based reward only for punishing spurious paths, but not for awarding paths with high reasonableness but low accuracy. That is, R_r merely serves as punishment when it is lower than R_s. Note that R_{min} is friendly to developers since it has no hyperparameters.

The other is the *weighted sum* of R_s and R_r:

$$R_{sum}(H, e_T) = \alpha R_r(H) + (1-\alpha) R_s(e_T), \quad (16)$$

where α decreases in the process of training. This equation is inspired by curriculum learning, which sets different optimizing goals in different learning stages (Bengio et al., 2009). We give priority to the path reasonableness in early epochs, forcing the agent to focus more on reasonable paths, and after we prefer the agent to concentrate on proper answers. Curriculum learning has the chance to achieve higher performance but makes it more difficult to tune parameters.

4.3 Overall Training Process

The overall training process of our method is shown in Figure 2. The agent starts from the query entity e_s and makes transitions according to its policy network. The history memory holds the history information of each past step. After the agent reaches the final entity e_T, we use Breadth-First Search to get all valid substitutions of its reasoning path. Then all substitutions as well as the reasoning path itself are used to calculate R_r and R_s which are combined together to get either R_{min} or R_{sum}.

Our PyTorch implementation and some pretrained models are released at `https://github.com/rubickkcibur/PSAgent`.

5 Experiment

We evaluate our model on four datasets and compare it with eight common baseline models. *Ours+min* and *Ours+sum* indicate our two approaches with R_{min} and R_{sum} respectively.

5.1 Setup

5.1.1 Datasets

We use five benchmark datasets for query answering: 1) UMLS (Kok and Domingos, 2007), 2) Kinship (Lin et al., 2018), 3) FB15K-237 (Toutanova et al., 2015), 4) WN18RR (Dettmers et al., 2018), and 5) NELL-995 (Xiong et al., 2017). Statistics are shown in Table 1.

Dataset	#Ent	#Rel	#Fact	#degree	
				avg.	median
Kinship	104	25	10,686	82.2	82
UMLS	135	46	6,529	38.6	28
FB15K-237	14,505	237	272,115	19.7	14
WN18RR	40,945	11	93,003	2.2	2
NELL-995	10,105	12	13,825	1.6	1

Table 1: Statistics of five KGs used in experiments.

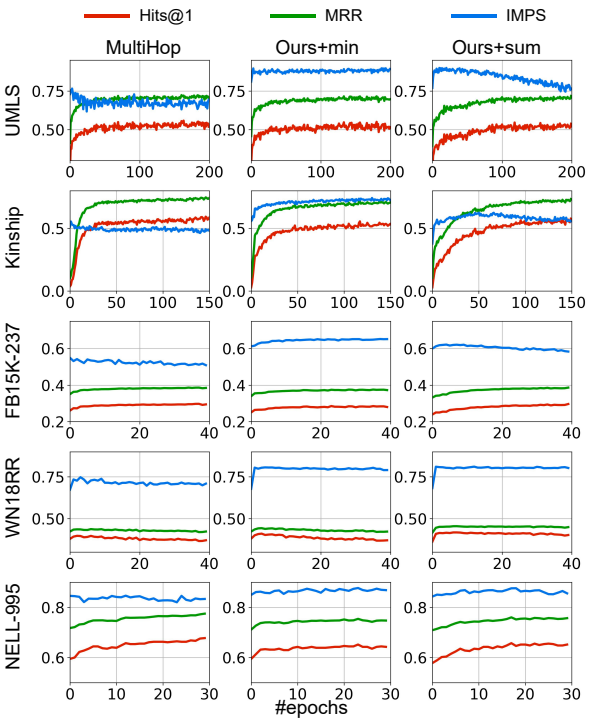

Figure 3: Hits@1 (red), MRR (green), and IMPS (blue) change over the validation set during training.

5.1.2 Baselines

We compare our method with five multi-hop reasoning models: 1) MINERVA (Das et al., 2018), the first end-to-end deep reinforcement learning model for multi-hop reasoning; 2) MultiHop (Lin et al., 2018), the first RL-based model incorporating embedding-based models as reward shaping function and proposing the action drop skill to mitigate spurious path problem; 3) MetaKGR (Lv et al., 2019), which leverages meta-information to improve reasoning performance on few-shot relations; 4) RARL (Hou et al., 2021), which utilizes mined logic rules to supervise the decision of the agent; 5) PAAR (Zhou et al., 2021), a fresh model com-

Dataset	UMLS				Kinship				FB15K-237				WN18RR				NELL-995			
	@1	@10	MRR	IMPS	@1	@10	MRR	IMPS	@1	@10	MRR	IMPS	@1	@10	MRR	IMPS	@1	@10	MRR	IMPS
DisMult	82.1	96.7	86.8	N/A	48.7	90.4	61.4	N/A	32.4	60.0	41.7	N/A	43.1	52.4	46.2	N/A	55.2	78.3	64.1	N/A
ComplEX	89.0	99.2	93.4	N/A	81.8	98.1	88.4	N/A	32.8	61.6	42.5	N/A	41.8	48.0	43.7	N/A	64.3	86.0	72.6	N/A
ConvE	93.2	99.4	95.7	N/A	79.7	98.1	87.1	N/A	34.1	62.2	43.5	N/A	40.3	54.0	44.9	N/A	67.8	88.6	76.1	N/A
MINERVA	78.5	97.4	85.9	68.6	59.6	95.8	73.0	49.7	26.0	40.7	31.3	57.5	43.1	52.9	46.0	71.9	51.6	79.6	62.3	81.4
MultiHop	**90.5**	**99.5**	94.2	65.3	78.8	**98.6**	86.8	48.4	**32.7**	56.4	40.7	50.6	41.2	52.0	45.1	73.9	65.0	82.9	71.9	79.1
MetaKGR	88.6	99.3	93.1	65.1	78.2	98.4	86.3	47.6	28.3	52.7	36.9	53.5	36.5	50.9	42.0	69.6	59.4	78.7	66.7	80.3
RARL	76.2	95.6	84.2	64.5	61.3	94.4	73.3	49.6	28.4	49.7	35.8	54.4	40.0	51.7	44.6	74.0	62.8	82.2	70.4	81.4
PAAR	89.5	**99.5**	94	67.2	72.7	96.3	81.4	47.5	32.1	55.0	40.0	55.3	40.6	**53.8**	44.9	73.8	65.2	83.9	72.2	77.3
Ours+min	89.1	98.9	93.2	**87.3**	76.2	98.1	85.0	**74.4**	30.9	56.1	39.5	**64.1**	42.1	50.9	45.1	**81.9**	**65.7**	**85.3**	**73.5**	**81.5**
Ours+sum	**90.5**	**99.5**	**94.6**	71.3	**79.1**	98.1	**86.9**	53.7	32.5	**57.0**	**40.9**	58.4	**43.4**	52.6	**46.8**	81.4	63.8	84.1	71.4	78.1
AMIE+	53.4	72.6	61.0	**92.1**	58.2	75.7	66.0	**76.7**	13.8	23.5	16.8	**68.8**	34.8	36.6	35.6	**99.2**	51.2	51.5	51.4	**95.1**

Table 2: QA Performance comparison on five datasets. The top part is embedding-based models and the bottom is RL-based models. *Ours+min* and *Ours+sum* denote our methods with R_{min} and R_{sum}, respectively. All metrics are multiplied by 100. IMPS is not applicable to embedding-based models because they lack reasoning paths.

bining hierarchical information in reward shaping for providing sufficient paths. Three embedding-based models are also picked up as comparisons: DistMult (Yang et al., 2015), ComplEX (Trouillon et al., 2016), and ConvE (Dettmers et al., 2018).

5.1.3 Hyperparameters

The entity embedding, relation embedding, and history embedding all have a size of 200. A three-layer LSTM is used for multi-hop reasoning models. The training batch size is 128. The maximum path length is 2 for UMLS and Kinship, and 3 for others. Following MultiHop (Lin et al., 2018), we use action drop in the training process, and the drop rate ranges from 0.1 to 0.9. For R_{sum}, there are two hyperparameters, decreasing interval D and decreasing rate η. We perform grid search on them and set $D = 10$ for UMLS and WN18RR, $D = 5$ for others, $\eta = 0.75$ for FB15K-237, and $\eta = 0.9$ for the other four datasets.

For MultiHop, MetaKGR, PAAR, and our method, we universally choose ConvE to implement the soft reward function. An entropy regularization term is added to the objective function in all RL-based models and the weight coefficient varies in (0, 0.1), as MINERVA does (Das et al., 2018).

We use Xavier initialization (Glorot and Bengio, 2010) to initialize parameters of embedding layers, and Adam optimizer (Kingma and Ba, 2015) to realize optimization where the learning rate is in (0.001, 0.003).

We perform beam search to get the final prediction and the beam size is 128 for all cases. A single training on NVIDIA Tesla V100 GPU costs 20 hours on FB15K-237, and at most 10 hours among all other datasets.

5.1.4 Evaluation Protocol

We choose Hits@k and Mean Reciprocal Rank (MRR) to evaluate the accuracy of predictions, and use Mean Path Spuriousness (MPS) to estimate the spuriousness of paths. For consistency, we use 1-MPS in measurement, denoted by IMPS.

For each test case $< e_s, r, e_o >$, the model takes as input the subject e_s and relation r, and returns a list of candidate answers $E_o = [e^1, e^2, \cdots, e^N]$ in decreasing order of confidence, as well as a list of corresponding reasoning paths $\mathcal{H}_o = [H^1, H^2, \cdots, H^N]$, where N is the beam size, and the termination of each path H^i is e^i for $i \in [1, N]$.

We use r_{e_o} to indicate the rank of e_o in E_o and H_o to represent the path along which the agent reaches e_o. Hits@k is the percentage of test cases where $r_{e_o} \leq k$, MRR is the mean of $1/r_{e_o}$, and MPS is the mean of $\mathrm{PS}(H_o)$.

5.2 Validation of the IMPS

To verify whether the IMPS metric (*i.e.* 1-MPS) could correctly evaluate the reasonableness of reasoning paths, we conduct experiments on rule-based reasoning models. Rule-based models mostly lack generalization but have highly credible results, since they use logical rules, either designed by experts or mined from datasets, to extract reasoning paths and infer new facts. That is to say, if the IMPS metric is a proper measurement of path reasonableness, rule-based models will perform beyond other models on it.

We choose AMIE+ (Galárraga et al., 2015) to mine rules and then infer answers following them. The results (the final bar of Table 2) meet our expectations. AMIE+ gets the highest IMPS scores on four datasets, exceeding all RL-based models. Compared to other RL-based ones, the gap between our model and AMIE+ is evidently smaller.

5.3 Model Comparison

Table 2 shows evaluation performance on query answering. Results of embedding-based models are quoted from MultiHop (Lin et al., 2018), and IMPS is not applicable to them because they have no reasoning paths. *Ours+min* and *Ours+sum* denote two variants of our approach with R_{min} and R_{sum}, respectively. For all approaches, We record performances on the validation set of each epoch during training and choose the best one (in terms of MRR) as the testing model.

In terms of Hits@k and MRR, embedding-based models generally perform better, while multi-hop approaches are comparable with embedding-based methods in some metrics, such as Hits@10 in UMLS, Hits@1 in Kinship and MRR in WN18RR. Among multi-hop reasoning models, *Ours+sum* outperforms previous ones in most cases.

As for IMPS metric, our methods largely surpass other RL-based models, Specifically, 27 percent in UMLS, 50 percent in Kinship, 10 percent in WN18RR, and 11 percent in FB15K-237. An exception is NELL-995, where most models behave nearly the same. It possibly results from the sparsity of NELL-995. Since the average node degree is lower than 2, most reasoning paths in NELL-995 may have few valid substitutions.

An interesting fact is that performance varies between our two methods. Generally, *Ours+min* gets higher IMPS scores, and *Ours+sum* behaves better on Hits@k and MRR. We believe that when the reasoning model's performance is not good enough, prediction accuracy and reasonableness keep consistent, but if higher performance is required, there is a trade-off between them. In most cases, reasonable paths lead to accurate answers, but in some special cases, the answer can not be accessed in a regular way. For example, the statement "*if A is metal, A is solid*" is reasonable for almost all metallic elements, but when A is mercury, following that will lead to a wrong answer. The little fallback in MRR of *Ours+min* may be caused by these special cases, and the more special cases in a dataset, the larger the gap is. *Ours+sum* formulation provides a manual way to make a balance, so we can get better MRR at the expense of IMPS.

5.4 Learning Process

We are interested in the effect on the dynamic learning process of three different rewards R_s, R_{min}, and R_{sum}. So we draw curves of Hits@1, MRR, and IMPS metrics on validation set during training. As shown in Figure 3, from left to right, the three columns are MultiHop, *Ours+min*, and *Ours+sum*, using R_s, R_{min}, and R_{sum} respectively.

In the early stage of training, the MultiHop's IMPS score falls as MRR and Hits@1 rise, which is much more significant on UMLS and Kinship. This phenomenon evidently shows the misleading of spurious paths. As a comparison, our two approaches have a higher IMPS score at the beginning and maintain or elevate it as MRR increases. As illustrated, the promotion of reasoning paths reasonableness by incorporating PS-aware reward (*i.e.* R_{min} and R_{sum}) is mainly reflected in the early training process.

Compared to *Ours+min*, the IMPS score of *Ours+sum* first lifts and then slowly decreases while the MRR score keeps rising. It justifies that prediction accuracy and path reasonableness are in consistency when performance is not sufficiently good, but need a trade-off if better performance is required. However, this balance is not obvious in WN18RR, where *Ours+sum* gets great scores in both MRR and IMPS with proper hyperparameters. We suppose one reason is that the proportion of special cases which can not be accessed by regular logical path is small.

MultiHop and *Ours+min* show a similar training convergence rate, while *Ours+sum* converges slower. The successive change of the coefficient α in R_{sum} makes the agent learn different goals and therefore hard to converge.

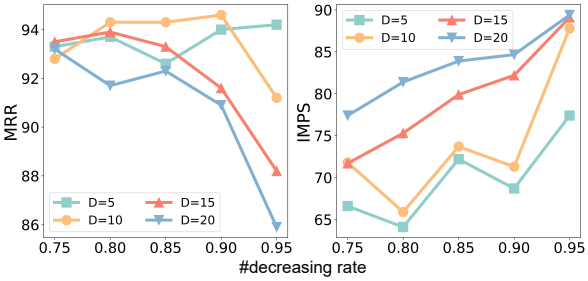

Figure 4: MRR (left) and IMPS (right) performance change w.r.t. decreasing interval D and rate η of R_{sum}. Each line represents metric variety over various decreasing rates for a certain decreasing interval.

5.5 Hyperparameter Study

We study the hyperparameters relevant to R_{sum}, specifically the decreasing interval D and decreasing rate η. Figure 4 shows MRR and IMPS results on UMLS with 20 permutations of D and η. We

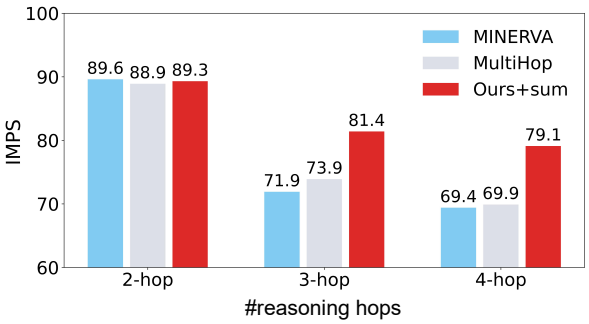

Figure 5: IMPS comparison among MINERVA (blue), MultiHop (gray), and our method (red), over different reasoning hops. IMPS metric is multiplied by 100.

figure out that when decreasing interval is small, MRR varies little with decreasing rate, and vice versa. It is quite fair because smaller D or η implies more priority to the accuracy-guided part of R_{sum} during training, which prompts better performance on MRR. However, settings of middle D and η achieve the best score, and we believe it reveals that proper tendency to the spuriousness-guided part of R_{sum} in the early training stage has a positive influence over accuracy. In terms of IMPS, it declines on the whole as η becomes smaller, and the larger D is, the slower it falls. Generally, in small and dense datasets, small decreasing intervals and rates would lead the agent to get higher scores on MRR, while larger intervals and rates make the agent prefer more reasonable paths. A balance where the agent gets high scores on both MRR and IMPS exists with refined parameters.

5.6 Analysis on Reasoning Hops

We also want to throw some light on the influence of the maximum length of reasoning paths (i.e., reasoning hops). We permute reasoning hops and test performance in IMPS of three models, MINERVA, MultiHop, and ours. We pick WN18RR as the benchmark because UMLS and Kinship are so dense that all entities can be reached within 2 hops starting anywhere. As Figure 5 shows, the reasonableness of paths picked by three models drops as reasoning hops increase, without exception. Compared to baseline models, our method has a better resistance against the tendency. Thereby, the gap between our method and baselines widens as reasoning hops go up.

6 Conclusion

In this paper, we discuss the spurious path problem which widely exists in the RL-based multi-hop reasoning models. To address this problem, we define a new metric named Path Spuriousness (PS) to quantitatively evaluate to what extent a path is spurious and consequently propose a new reward that considers both the prediction accuracy and path reasonableness. Under the guidance of the reward, the agent can be aware of not only whether its prediction is right, but also the spuriousness of its reasoning path, and thus avoid spurious paths.

Experiments show our method largely outperforms baseline models in terms of PS, and keep comparable to the state-of-the-art performance of prediction accuracy. Detailed analysis indicates that a trade-off between pursuing better prediction accuracy and keeping high path reasonableness exists. Its significance varies among different datasets. We provide a method to make a balance by manually pruning hyperparameters.

Analysis on reasoning hops shows potential in long-hop reasoning tasks. In future work, we would like to further investigate it.

Limitations

Our job has two major limitations regarding the definition of PS and the computational cost.

The definition of PS (*i.e.*, Equation 10 and Equation 11) is constructed on the assumption that all facts in a KG are correct. However, some datasets contain mistaken facts in practice. It could make the PS a biased estimation. Moreover, experiments on NELL-995 indicate that the sparsity of KG may limit the effectiveness of our PS-aware reward. How to avoid spurious paths in sparse KGs and even KGs that contain mistakes remains a hard topic for future work.

The other deficiency is that we use a breadth-first tree search to find path substitutions, whose worst-case time cost increases exponentially as the search depth grows. This shortcoming has been reflected by the experiment time cost on FB15K-237 and limits our method's capability to scale to much larger datasets. We expect pruning skills or end-to-end deep models would be able to tackle this problem and leave it for future work.

Acknowledgements

This work is supported by National Key R&D Program of China (2021ZD0113903). We also thanks for the computing infrastructure provided by Beijing Advanced Innovation Center for Big Data and Brain Computing.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*, pages 2787–2795.

Sergey Brin. 1998. The pagerank citation ranking: bringing order to the web. *Proceedings of ASIS, 1998*, 98:161–172.

Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.*, 141.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *ICLR*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*, pages 1811–1818.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2015. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB J.*, 24(6):707–730.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256.

Kelvin Guu, Panupong Pasupat, Evan Zheran Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. In *ACL*, pages 1051–1062.

Yu-Jung Heo, Eun-Sol Kim, Woo Suk Choi, and Byoung-Tak Zhang. 2022. Hypergraph transformer: Weakly-supervised multi-hop reasoning for knowledge-based visual question answering. In *ACL*, pages 373–390.

Marcel Hildebrandt, Jorge Andres Quintero Serna, Yunpu Ma, Martin Ringsquandl, Mitchell Joblin, and Volker Tresp. 2020. Reasoning on knowledge graphs with debate dynamics. In *AAAI*, pages 4123–4131.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Zhongni Hou, Xiaolong Jin, Zixuan Li, and Long Bai. 2021. Rule-aware reinforcement learning for knowledge graph reasoning. In *ACL Findings*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4687–4692.

Jiacheng Huang, Yao Zhao, Wei Hu, Zhen Ning, Qijin Chen, Xiaoxia Qiu, Chengfu Huo, and Weijun Ren. 2022. Trustworthy knowledge graph completion based on multi-sourced noisy data. In *WWW*, pages 956–965.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *ACL*, pages 687–696.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Stanley Kok and Pedro M. Domingos. 2007. Statistical predicate invention. In *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 433–440.

Deren Lei, Gangrong Jiang, Xiaotao Gu, Kexuan Sun, Yuning Mao, and Xiang Ren. 2020. Learning collaborative agents with rule guidance for knowledge graph reasoning. pages 8541–8547.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *EMNLP*, pages 3243–3253.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 2168–2178.

Xin Lv, Yuxian Gu, Xu Han, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2019. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. pages 3374–3379.

Wenchang Ma, Ryuichi Takanobu, and Minlie Huang. 2021. Cr-walker: Tree-structured graph reasoning and dialog acts for conversational recommendation. In *EMNLP*, pages 1839–1851.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814.

Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics.

Ribana Roscher, Bastian Bohn, Marco F. Duarte, and Jochen Garcke. 2020. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216.

Thomas Pellissier Tanon, Daria Stepanova, Simon Razniewski, Paramita Mirza, and Gerhard Weikum. 2017. Completeness-aware rule learning from knowledge graphs. In *ISWC*, volume 10587 of *Lecture Notes in Computer Science*, pages 507–525.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509. The Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080.

Guojia Wan and Bo Du. 2021. Gaussianpath: A bayesian multi-hop reasoning framework for knowledge graph reasoning. In *AAAI*, pages 4393–4401.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *EMNLP*, pages 564–573.

Weiwen Xu, Yang Deng, Huihui Zhang, Deng Cai, and Wai Lam. 2021. Exploiting reasoning chains for multi-hop science question answering. In *EMNLP*, pages 1143–1156.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

Yongqi Zhang and Quanming Yao. 2022. Knowledge graph reasoning with relational digraph. In *WWW*, pages 912–924.

Zhanqiu Zhang, Jie Wang, Jieping Ye, and Feng Wu. 2022. Rethinking graph convolutional networks in knowledge graph completion. In *WWW*, pages 798–807.

Xingchen Zhou, Peng Wang, Qiqing Luo, and Zhe Pan. 2021. Multi-hop knowledge graph reasoning based on hyperbolic knowledge graph embedding and reinforcement learning. In *IJCKG*, pages 1–9.

A Experiment Details

A.1 Case Study

We select four test cases in the UMLS dataset and look deep into the difference between reasoning paths extracted by MultiHop and our method. Table 3 shows the comparisons. In each case, we only concentrate on the accurate predictions as well as their paths. The reasonableness of paths is indicated by $R_r(H_o)$, and the rank of correct answer e_o is represented as r_{e_o}.

For most cases, the reasoning path of our method to obtain the correct answer is much more reasonable than MultiHop.

(1) Case 1. In case 1, to get the relation *location_of*, our reasoning path through relations *adjacent_to* and *location_of* is much more convinced than MultiHop's path through relations *issue_in* and *method_of*$^{-1}$. That's because *Adjacent* relation is clearly more relevant than *method* relation since we want to know where is the location.

(2) Case 2. Case 2 shows insight into the different preferences between the two approaches in respect of picking paths. The two paths differ in the last relation, where MultiHop reaches the final answer via *issue_in*$^{-1}$ while our method picks *issue_in*.

Picking *issue_in* is reasonable in this case.

(a) It is a special coincidence that *occupation_or_discipline* and *biomedical_occupation_or_discipline* are reciprocal causation in UMLS dataset. However, in terms of reasonableness, the fact "*A causes C* " cannot be derived by "*A causes B*" and "*C causes B*" (i.e., *issue_in(A, B)* $\wedge$ *issue_in*$^{-1}$*(B, C)*).

(b) Guided by the embedding-based model, MultiHop only captures the semantic information between *occupation_or_discipline* and *biomedical_occupation_or_discipline* and obtains a correct answer incidentally. In contrast, our method not only concentrates on the semantic correctness of certain entities but also takes path spuriousness into account. The extremely lower path reasonableness of *issue_in*$^{-1}$ prohibits our method from picking it to construct the reasoning path.

(3) Case 3. Though the path returned by MultiHop gets a score 0.49 of R_r, it is still lower than the path given by our method, considering that adjacent things possibly belong to different concepts, like lung and air. Relation *part_of* is definitely more proper to derive an *conceptual_part_of*relation.

(4) Case 4. In case 4, we get a reasonable path

Case 1:<*body_location_or_region, location_of, therapeutic_or_preventive_procedure*>		r_{e_o}	$R_r(H_o)$
MultiHop	*body_location_or_region* $\xrightarrow{issue\_in}$ *occupation_or_discipline* $\xrightarrow{method\_of}^{-1}$ *therapeutic_or_preventive_procedure*	3	0.22
Ours+min	*body_location_or_region* $\xrightarrow{adjacent\_to}$ *body_part_organ_or_organ_component* $\xrightarrow{location\_of}$ *therapeutic_or_preventive_procedure*	0	0.71
Case 2:<*physical_object, issue_in, biomedical_occupation_or_discipline*>		r_{e_o}	$R_r(H_o)$
MultiHop	*physical_object* $\xrightarrow{issue\_in}$ *occupation_or_discipline* $\xrightarrow{issue\_in}^{-1}$ *biomedical_occupation_or_discipline*	0	0.01
Ours+min	*physical_object* $\xrightarrow{issue\_in}$ *occupation_or_discipline* $\xrightarrow{issue\_in}$ *biomedical_occupation_or_discipline*	0	0.99
Case 3:<*cell_component, conceptual_part_of, body_system*>		r_{e_o}	$R_r(H_o)$
MultiHop	*cell_component* $\xrightarrow{adjacent\_to}$ *body_space_or_junction* $\xrightarrow{conceptual\_part\_of}$ *body_system*	3	0.49
Ours+min	*cell_component* $\xrightarrow{part\_of}$ *cell* $\xrightarrow{conceptual\_part\_of}$ *body_system*	0	0.99
Case 4:<*indicator_reagent_or_diagnostic_aid, interacts_with, chemical*>		r_{e_o}	$R_r(H_o)$
MultiHop	*indicator_reagent_or_diagnostic_aid* $\xrightarrow{causes}$ *experimental_model_of_disease* $\xrightarrow{inv\_causes}$ *chemical*	0	0.05
Ours+min	*indicator_reagent_or_diagnostic_aid* $\xrightarrow{interacts\_with}$ *hazardous_or_poisonous_substance* $\xrightarrow{interacts\_with}$ *chemical*	5	0.99

Table 3: Comparison of selected paths between MultiHop and our method on 4 cases. The reasonableness of paths is indicated by $R_r(H_o)$ and r_{e_o} represents the rank of correct answer e_o in candidate list E_o.

Dataset	HC_{min}	PCA_{min}	#Rules
UMLS	0.50	0.40	160
Kinship	0.60	0.40	74
WN18RR	0.10	0.50	34
NELL-995	0.10	0.30	8
FB15K-237	0.30	0.24	1074

Table 4: AMIE+ setting details. HC_{min} and PCA_{min} indicate the minimum head coverage and PCA confidence respectively, which is threshold for mining rules.

but with a lower rank. That is, the agent does not always pick a path with a high R_r. It may be limited to the representation power of the policy network. We leave this problem for future research.

A.2 AMIE+ Settings

To validate our proposed metric IMPS, we choose AMIE+ to mine rules and then infer answers following them. There are two hyperparameters of AMIE+: the head coverage and PCA confidence (denoted by HC_{min} and PCA_{min} respectively). The higher the two parameters are, the smaller the number of qualified mined rules. In Table 4 we show the settings of HC_{min} and PCA_{min}, and the number of qualified rules we mined using AMIE+.