

Article

Causal Reinforcement Learning for Knowledge Graph Reasoning

Dezhi Li, Yunjun Lu *, Jianping Wu, Wenlu Zhou and Guangjun Zeng

School of Information and Communication, National University of Defense Technology, Wuhan 430019, China; ldzfreedom@sina.com (D.L.); wjp860343510@sina.com (J.W.); zhouwenlu17@nudt.edu.cn (W.Z.); guangjun_zeng@nudt.edu.cn (G.Z.)

* Correspondence: luyunjun@nudt.edu.cn

Abstract: Knowledge graph reasoning can deduce new facts and relationships, which is an important research direction of knowledge graphs. Most of the existing methods are based on end-to-end reasoning which cannot effectively use the knowledge graph, so consequently the performance of the method still needs to be improved. Therefore, we combine causal inference with reinforcement learning and propose a new framework for knowledge graph reasoning. By combining the counterfactual method in causal inference, our method can obtain more information as prior knowledge and integrate it into the control strategy in the reinforcement model. The proposed method mainly includes the steps of relationship importance identification, reinforcement learning framework design, policy network design, and the training and testing of the causal reinforcement learning model. Specifically, a prior knowledge table is first constructed to indicate which relationship is more important for the problem to be queried; secondly, designing state space, optimization, action space, state transition and reward, respectively, is undertaken; then, the standard value is set and compared with the weight value of each candidate edge, and an action strategy is selected according to the comparison result through prior knowledge or neural network; finally, the parameters of the reinforcement learning model are determined through training and testing. We used four datasets to compare our method to the baseline method and conducted ablation experiments. On dataset NELL-995 and FB15k-237, the experimental results show that the MAP scores of our method are 87.8 and 45.2, and the optimal performance is achieved.

Keywords: Priori knowledge; counterfactual; reinforcement learning; knowledge graph reasoning



Citation: Li, D.; Lu, Y.; Wu, J.; Zhou, W.; Zeng, G. Causal Reinforcement Learning for Knowledge Graph Reasoning. *Appl. Sci.* **2024**, *14*, 2498. <https://doi.org/10.3390/app14062498>

Academic Editor: Andrea Prati

Received: 14 February 2024

Revised: 7 March 2024

Accepted: 14 March 2024

Published: 15 March 2024



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

1. Introduction

In recent years, knowledge graphs have been widely used in several fields. However, for various professional fields, the constructed knowledge graphs are usually sparse and the implicit relationship between entities has not been fully examined, resulting in a poor practical application effect. Therefore, it is necessary to complete the knowledge graphs; the typical method is knowledge graph reasoning.

Knowledge graph reasoning tasks include Knowledge Graph Completion (KGC), quality verification, path prediction, relationship reasoning, and conflict detection [1]. This paper mainly focuses on KGC and transforms the knowledge graph reasoning problem into a problem predicting the missing part of the triple. For example, as depicted in Figure 1, we obtain two paths based on the existing knowledge graph: “Christopher Nolan^{Born in}London” and “London^{Located in}Britain”. Our task is to complete the triple: (Christopher Nolan, nationality, ?). Since we do not have the path “Christopher Nolan^{Nationality}British”, we use the dotted line to represent the edge of the relationship. From knowledge graph reasoning, the above two paths are equivalent to “Christopher Nolan^{Nationality}British”. So, we complete the triple: (Christopher Nolan, nationality, British).

In the above reasoning process, we start from the Nolan node. For the relationship of “nationality”, the association degree of the edges linked to the Nolan node differs,

highlighting that “Born in” is more relevant to finding a relationship of nationality than “Direct” and “Partner”. This is because humans have prior knowledge that “place of birth” is more likely to determine “nationality” than “director” and “partner”.

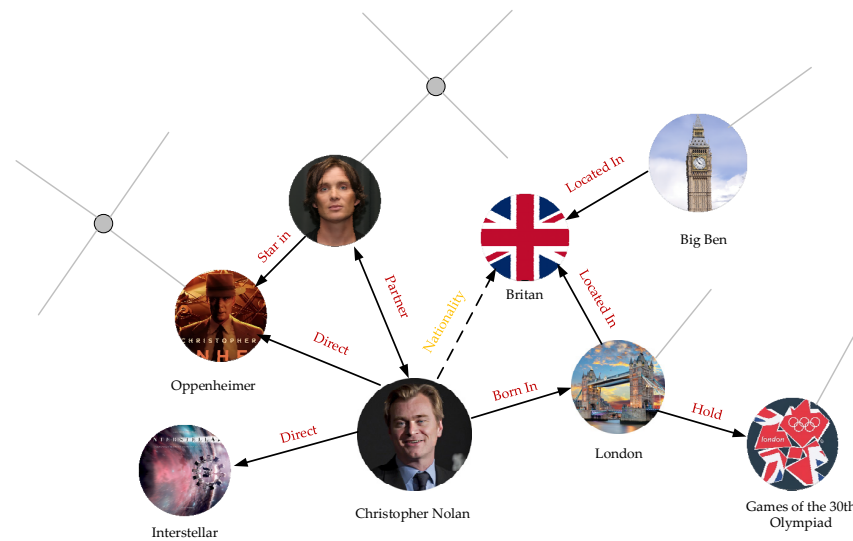


Figure 1. Schematic diagram of knowledge graph reasoning. The dotted lines represent relational paths that were unknown before reasoning.

Reinforcement learning is applied to knowledge graph reasoning tasks owing to its adaptability to complex environments [2]. Reinforcement learning regards the above reasoning process as the following problem: finding a path connecting the known head entity and tail entity, modeling this problem as a sequential decision problem, and solving it using the method based on policy gradient [3]. Since the existing reinforcement learning models do not conduct in-depth research on the knowledge graph, this paper employs causal inference to analyze the knowledge graph and obtain prior knowledge, thereby improving inference performance. For instance, in Figure 1, we want to complete the triple (person, language, ?), and there are two paths in the knowledge graph: “person $\xrightarrow{\text{Star in}}$ film $\xrightarrow{\text{Country}}$ Britain” and “person $\xrightarrow{\text{Star in}}$ film $\xrightarrow{\text{Language}}$ English”. We obtain prior knowledge through causal inference, i.e., the relationship “language” in the second path is more related to the relationship in the triple to be completed than the relationship “country” in the first path. So, we obtain the importance of the relationship in the path and thus guide the path selection.

The major contributions of this work are as follows:

1. We propose a prior knowledge generation method based on causal inference. Specifically, it is implemented through the relationship importance identification module. The generated prior knowledge is mainly used to measure the contribution of the relationship in the triple to be completed.
2. We introduce a new method combining causal inference and reinforcement learning that applies to knowledge graph reasoning. Specifically, the prior knowledge is integrated into the control strategy of the reinforcement model, allowing the agent to select the relation more accurately in each step. Our model also applies to large-scale knowledge graphs.
3. The experiments demonstrate that our method’s performance outperforms the current baseline methods in most cases. Additionally, we conduct ablation experiments to highlight that our method is more effective than solely employing reinforcement learning.

The remainder of this paper is organized as follows. Section 2 briefly introduces the related work. Section 3 presents the methodological details, and Section 4 conducts experiments and analyzes the experimental results. Section 5 summarizes this work and provides future research directions.

2. Related Work

2.1. Knowledge Graph Reasoning

The existing methods in the field of knowledge graph reasoning involve two categories: methods based on distributed representation and based on relational path [4]. Knowledge graph methods based on distributed representation predict the missing tail entities by learning the low-dimensional embedded representation of known entities and relationships in triples. The most common method is TransE, which was proposed by Bordes et al. [5]. Since the TransE method can only deal with the one-to-one relationship problem, many subsequent methods evolved from it. For instance, Wang et al. [6] and Lin et al. [7] introduced hyperplanes and mapping matrices to deal with the many-to-one relationship problem and proposed the TransH method and TransR method, respectively. In addition, Ji et al. [8] introduced a dynamic matrix and proposed the TransD method to solve this problem. However, these improved models cannot model and simultaneously infer all kinds of relationships (symmetry and composition). Thus, Pasquale et al. [9] developed ConvE, a two-dimensional convolutional link prediction method. Furthermore, Trouillon et al. [10] proposed the ComplEx method, which introduced complex number space into knowledge representation. The model structure of ComplEx is relatively simple, since only Hermitian dot products are used.

The above methods cannot solve the problem of multi-step knowledge reasoning, and therefore, solutions based on the relational path have been developed. Such a traditional model is the PRA (Path Ranking Algorithm) [11], characterized by strong interpretability and automatic rules discovery. However, this model is ineffective in dealing with low-frequency relationships and low-connectivity graphs (sparse data). In addition, when the graph is large enough, path extraction is time-consuming. Thus, Xiong et al. [12] and Das et al. [13] introduced reinforcement learning into knowledge graph reasoning, considering the relational path selection problem as a Markov decision process with better reasoning effect and interpretability. On this basis, Wan et al. [14] proposed an improved multi-level reinforcement learning model, which encodes historical knowledge and action space, achieving appealing results. The attention mechanism introduced by Wang et al. [15] enables the reasoning process to be memorized, and the graph neural network can enhance the entity semantics. It should be noted that the models introduced above do not fully employ the knowledge graph. Therefore, the accuracy and efficiency of their methods can be further improved.

2.2. Casual Inference

Causal inference mainly refers to inferring the relationship between the cause-and-effect variables. Specifically, appropriate intervention on variable X will lead to a change in the distribution of variable Y , but intervention on Y will not lead to a change in X . In this case, X is the cause variable of Y , and Y is the effect variable of X . A causal inference model mainly solves two kinds of problems. One is the intervention problem, i.e., the variable is set to a certain value, and the causal relationship is analyzed according to the data distribution change. The other problem is the counterfactual problem, i.e., the probability that events A and B do not occur is inferred when events A and B occur [16–20]. Since the problem of relational path selection in knowledge graphs can be transformed into a counterfactual problem, this paper focuses on counterfactual problems [21].

Counterfactual reasoning generally comprises three steps: (1) Traceability: Updating the value of a noise term using evidence. This step accounts for a particular noise with evidence in mind; (2) Intervention: Modifying the original model, replacing the structural equation with counterfactual variables, and obtaining the modified model; (3) Prediction: The modified model and noise are used to re-estimate the outcome variables, and the counterfactual results are obtained [22]. Some studies investigate the counterfactual problem, with counterfactual reasoning improving the reinforcement learning algorithms' sample efficiency and interpretability [23]. For instance, Madumal et al. [24] proposed a behavioral influence model based on a structural causal model, which used the causal model for coun-

terfactual analysis to improve the interpretability of the model. Lu et al. [25] developed a counterfactual-based data enhancement algorithm, which used the Structural Causal Model (SCM) to model the environment dynamics and estimate the causal effect based on the commonalities and differences in multi-domain data. Buesing et al. [26] introduced a CounterFactually Guided Policy Search algorithm CF-GPS, which counterfactually evaluates any policy based on SCM to improve policy performance and eliminate the model prediction bias. However, there are few studies on the combination of counterfactual problems and knowledge graph, and how counterfactual problems will help knowledge graph research remains to be explored.

2.3. Reinforcement Learning

Reinforcement learning is an important method in behavioral decision-making and control [27]. Specifically, the agent takes action at any time according to the next step's current status and reward [28]. The way an agent chooses an action is called a policy, and reinforcement learning maximizes the cumulative reward by constantly improving the policy.

Reinforcement learning algorithms can be divided into model-based and model-free methods depending on whether the agent uses the dynamic model of the environment in policy updating. In the model-free reinforcement-learning method, the agent directly interacts with the environment and optimizes the policy end-to-end, which is easier to implement and has better asymptotic performance. Classic algorithms include SARSA (State-Action-Reward-State-Action) [29], Q-learning [30], deep Q-networks [31,32], and their variants [33]. Many studies have recently applied reinforcement learning to knowledge graph reasoning [34,35]. Indeed, Lin et al. [36] extended MINERVA and proposed the MultiHopKG model to solve the above problems. Wang et al. [14] proposed AttnPath to quantify the difficulty of relation learning through MSR and MRR and redesigned the model's control strategy. At the same time, to avoid the stagnation of agents on the same entity, AttnPath designs a new learning mechanism that enables the agent to move forward at every step. Shen et al. [15] solved the problem of reward sparsity using a Monte Carlo search to improve the efficiency of knowledge reasoning. However, these models are highly data-dependent and perform poorly in non-stationary and heterogeneous scenarios.

2.4. Causal Reinforcement Learning

Causal reinforcement learning combines causal inference theory and reinforcement learning, allowing agents to understand the causal relationship and make correct decisions. Additional assumptions or prior knowledge is often required to achieve the above goals. Specifically, causal inference empowered reinforcement learning using improving sample efficiency, enhancing the generalization ability of agents, and reducing the impact of correlation on agents [37].

Maximilian et al. [38] measured the causal effect of an action on an object through constructing a causal quantity, thereby guiding the agent to explore the environment more effectively. Jiaxian et al. [39] enhanced the model's generalization ability through measuring the controlled direct effect and reducing the encoding of the state irrelevant to the action into the context information. Pim et al. [40] believed that more information is not necessarily better, revealing the false causality's misleading effect on reinforcement learning control strategy. Zhihong et al. [41] investigated the problem of false correlation in offline reinforcement learning and avoided uncertainty by pessimistic criteria to neglect the influence of false correlation on the model control strategy.

3. Proposed Method

This section describes the proposed model. First, the KGC problem is defined in detail. Then, the general framework of the proposed method is described, as well as how to extract prior knowledge through the causal inference module. Finally, the reinforcement learning framework is introduced.

3.1. Problem Definition

Let H denote the set of head entities, T is the entity found in the triple, and R represents the relation in the triple. G represents a known knowledge graph, $G = \{(h_i, r_i, t_i)\} \subseteq \{H, R, T\}$, $h_i \in H, r_i \in R, t_i \in T, i \in (1, n)$, where i represents the number of steps the agent moved on the knowledge graph with h_q as the starting point and the target entity as the endpoint. Our mission is to find the missing tail entity t_q given the triple $(h_q, r_q, ?)$. The proposed method solves the KGC problem based on a relational path. Therefore, we must find a path to the target tail entity in the knowledge graph, i.e., our goal is to find such a path denoted $p = \{h_q, r_1, r_2, \dots, r_n, t_q\}$. This goal is equivalent to finding a path with semantics equal to r_q .

3.2. Overall Architecture

Figure 2 illustrates our method’s framework. First, according to the triple $(h_q, r_q, ?)$ and the path extracted from the knowledge graph, the prior knowledge is obtained through the counterfactual theory in causal inference. The prior knowledge is represented using the relationship weight in each path relative to the relationship to be queried and obtained using the relationship importance identification module. The relationship importance identification module will be described in detail in the next section. After obtaining the prior knowledge, we combine it with the policy network to guide the agent to make the right choice. The policy network mainly uses a Long Short-Term Memory (LSTM) to represent semantic information, which will be described later. Then, the reasoning step and the tail entity to be queried will be obtained, and finally, the knowledge graph will be updated.

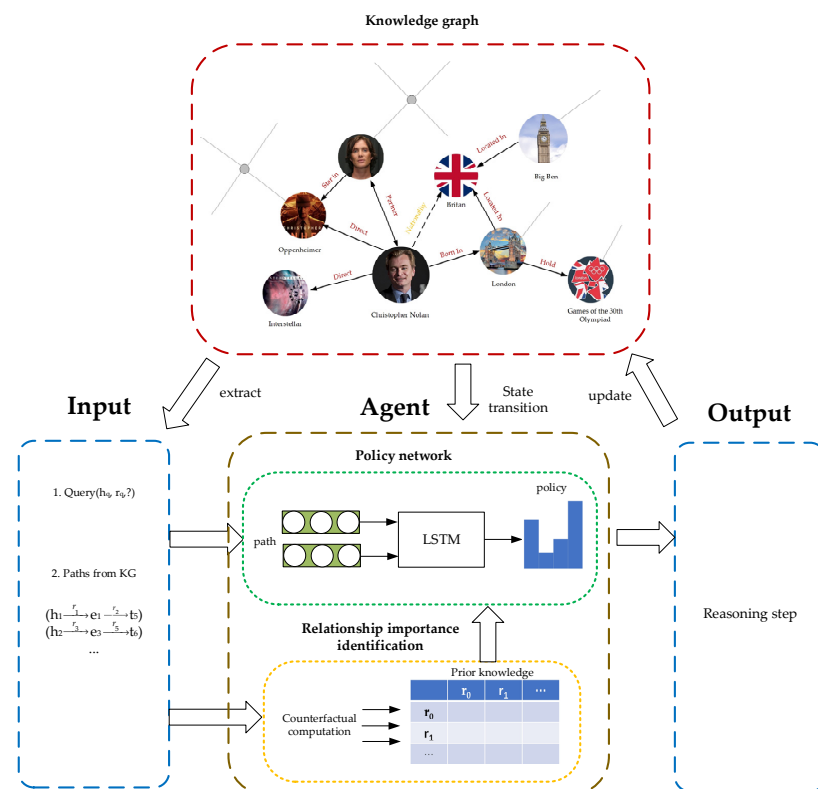


Figure 2. Overall framework of the casual reinforcement learning model. Combined with the question to be queried, starting from the target entity, the possible paths are extracted from the knowledge graph. In each step of the agent movement, the prior knowledge is obtained through the relation importance identification module, the reinforcement learning policy network is adjusted, and the state after the movement is updated. The agent updated the knowledge graph after reaching the target entity.

3.3. Relationship Importance Identification

Acquiring prior knowledge from the knowledge graph can assist in efficiently selecting the path closest to the real answer from the relational paths and better completing the missing tail entity. We introduced that prior knowledge is represented by the weight of the relationship in each path relative to the relationship to be queried. In this way, we quantify the importance of each relational path in the knowledge graph. This weight is implemented through a counterfactual question in causal inference. Specifically, for triple (h_q, r_q, t_q) with multiple paths from h_q to t_q , we obtain the following conclusion through the counterfactual definition: when the relation r_q changes, the tail entity will also change.

For the example in Figure 3, for the relationship R_0 , if we change r_2 to r_4 in the path, and the rest remains unchanged, the relationship R_0 changes to R_6 . Thus, we can obtain prior knowledge, as r_2 is more important for the relationship R_0 . Then, we record this prior knowledge and the corresponding relationship individually.

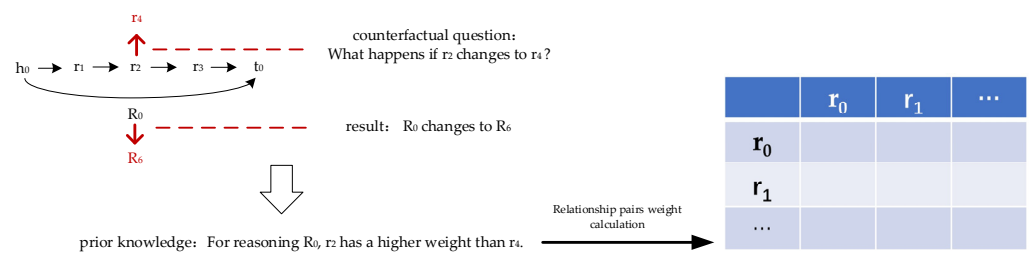


Figure 3. Relationship importance identification module based on causal inference.

Next, we construct a table of prior knowledge to represent the importance of the relationship. We denote this table by f , where r_i represents the relationship between the currently pointed entity and the next-hop entity after the agent moved i steps and the number of times we obtain the prior knowledge that “ r_i is important for r_q ” by $f[r_q][r_i]$. Thus, the weight of r_i to r_q is:

$$\omega_R[r_q][r_i] = \frac{f[r_q][r_i]}{\sum_{r_i \in R} f[r_q][r_i]} \quad (1)$$

where ω_R denotes the set of weights and the set of extracted prior knowledge. A larger weight of r_i indicates that r_i should be selected first when selecting a path.

Similarly, we use ω_E to represent the weight of the entity:

$$\omega_E[r_q][e_i] = \{ \omega_R[r_q][r_j] | (e_i, r_j, e_k) \in G, e_k \in T \} \quad (2)$$

where $\omega_E[r_q][e_i]$ is the importance of the edge from entity e_i concerning relation r_q .

Figure 3 depicts the structure of the relationship importance identification module based on causal inference. Since we have obtained the knowledge graph, it is easier to obtain prior knowledge through counterfactual methods in causal inference. In contrast to graphical models that require negative sampling, the proposed model only requires the correct triples, thus avoiding the noise introduced by false negative samples due to negative sampling.

3.4. Reinforcement Learning Framework

This section describes how to design a reinforcement learning model with the relationship importance identification module. Compared with the traditional reinforcement learning framework, we introduce prior knowledge to enhance the performance of our method. Specifically, we transform the path selection problem into a partially observable Markov decision process. Suppose the following triple $(h_q, r_q, ?)$ is looked up in the knowledge graph. The agent starts from h_q and chooses a relation according to the inference problem, thus jumping to the new entity. Based on the new entity jumped to, we continue

to select the relationship and iterate several steps until we find the ultimate goal. In this process, the agent prefers the most promising relationship for state transition according to the optimization process and updates the policy network through the reward function to find the same correct path as the r_q semantics.

3.4.1. Environment

Next, we represent the environment using five elements: (S, O, A, P, R).

S (State). The agent moves until it finds the target entity, so the state space at time t can be expressed in the following terms. One is the head entity h_q , which is the starting position of the agent. The second is the relation r_q , which is known in triples, and it is necessary to find a path in the knowledge map that is semantically equal to the relation r_q . The tail entity t_q is the target to be queried, which can test the accuracy of the inference algorithm. The third is entity e_t , representing the entity the agent is pointing to. According to the description of the above three parts, we can express the state space as: $s_t = (h_q, r_q, t_q, e_t)$.

O (Optimization). The optimization aims to quantify the prior knowledge obtained using the relationship importance identification module, i.e., to express the importance of the relation as a weight value. Taking entity e_t as the starting point, the prior knowledge is expressed as $\omega_E[r_q]$, and the optimization function is defined as $o_t = O(s_t = (h_q, r_q, t_q, e_t)) = (h_q, r_q, e_t, \omega_E[r_q])$.

A (Actions). The action space is the set of all edges from the current entity to the next entity. Therefore, the action space can be expressed as $A_t = \{(r_{t+1}, e_{t+1}) | (e_t, r_{t+1}, e_{t+1}) \in G\}$. Based on this, we represent the case where the inference terminates, which can be represented using an edge pointing to the entity.

P (Transition). The state transition space refers to the transition of the relationship path between the agent and the head entity and the transition of the current entity after the agent moves from one entity to another. The target entity to be searched remains unchanged during the agent's movement. Thus, the state transition space can be expressed as: $P(S_t, A_t) = P((h_q, r_q, t_q, e_t), (r_{t+1}, e_{t+1})) = (h_q, r_q, t_q, e_{t+1})$.

R (Rewards). Agents must constantly select actions from the action space, move forward in the knowledge map, and finally find the target entity. Therefore, we define the reward in the following form:

$$R = \begin{cases} 1 & \text{if } e_T = t_q \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where T denotes the last time the agent moves on the knowledge graph.

3.4.2. Policy Network

We use the policy network to guide the agent to make the right choice in the action space. The policy network is constructed using the prior knowledge extracted using causal inference and the neural network. Specifically, we first set a standard value φ , while the current time is t , and the entity pointed by the agent is e_t . The agent obtains the weight value of each edge from the optimization space and compares it with φ . If $\max(\omega_E[r_q][e_t])$ is larger than φ , the edge with the largest value is selected from the optimization space. Otherwise, the corresponding edge is selected according to the reinforcement learning model.

When the neural network selects an action, a path memory component (LSTM) is considered to encode the path information so that the history exploration is memorized [42]. History embedding h_t represents encoding the path that the agent moves on the knowledge graph before time t . The path is the composition of a series of entities and relationships, which can be represented as $\{h_q, r_1, e_1, r_2, e_2, \dots, r_t, e_t\}$. Then, h_q, r_q , and e_t are concatenated and input into the nonlinear Multi-Layer Perceptron (MLP) [21]. The policy network π is defined as follows:

$$\pi(r_t|r_q, e_t, h_t, h_q) = \begin{cases} \omega_E[r_q][e_t] & \text{if } \max(\omega_E[r_q][e_t]) \geq \varphi \\ \text{soft max}(A_t(W_2 \text{RELU}(W_1[h_t; h_q; r_q; e_t; a_t]))) & \text{otherwise} \end{cases} \quad (4)$$

where W_1 and W_2 are weighting matrices.

3.4.3. Training

During training, according to the above strategy network, we update the parameter φ to maximize the expected cumulative reward J :

$$J(\varphi) = E_{(e_1, r, e_2) \sim D} E_{a_1, a_2, \dots, a_{T-1} \sim \pi} [R(s_T | h_q, r_q)] \quad (5)$$

where D represents all real and observed triples, the first E is the empirical average over the training dataset, and s_T is the entity to which the agent is ultimately directed. The gradient update formula of the policy network can be expressed as follows:

$$\frac{\partial J(\varphi)}{\partial \varphi} = \frac{\partial}{\partial \varphi} \log \pi(r_t | r_q, e_t, h_t, h_q) \quad (6)$$

The specific training steps are described in Algorithm 1. First, the parameter θ (Line 01) is initialized. Next, for each episode (Line 02), a path p (Line 03) is created, and the history embedding h_0 is initialized to 0 (Line 04). Next, for the historical path of the current node (Line 05), the agent calculates the history embedding, trains the policy network, and obtains the next action vector (Line 06 and Line 07). Next, the environment will transition to the next state and reward the agent for its choice (Line 08). By continuously using the formula to update φ (Line 09), the relationship path is determined to obtain the location of the target entity.

Algorithm 1. Training Procedure

```

01 Initialize  $\varphi$ 
02 for episode  $\leftarrow 1$  to  $N$  do
03   Initialize path  $p$ 
04   Initialize entity representation  $h_0$ 
05   for  $j \leftarrow 1$  to  $|p|$  do
06     Calculate history embedding  $h_t \leftarrow \text{LSTM}(h_{t-1}, [a_{t-1}; r_q; h_q; e_t])$ 
07     Obtain reward  $R(p)$ 
08      $\nabla J \leftarrow \frac{\partial}{\partial \varphi} \log \pi(r_t | r_q, e_t, h_t, h_q)$ 
09     Update  $\varphi$ 
10   end for
11 end for

```

4. Experiments

4.1. Datasets

This work employs several classes of data sets that are widely used in knowledge graph reasoning, including (1) FB15K-237 [43], (2) WN18RR [44], (3) NEL-995 [45], (4) YAGO3-10 [46]. Table 1 reports the details of the four datasets.

Table 1. Detailed of datasets FB15k-237, WN18RR, YAGO3-10 and NELL-995.

Dataset	E	R	Training Triples	Test Triples
FB15K-237	14,541	237	272,115	20,466
WN18RR	40,943	11	86,835	3134
NELL-995	75,492	200	154,213	3992
YAGO3-10	123,182	37	1,079,040	5000

FB15K-237 is a subset of FB15K, while FB15K is derived from the Freebase dataset. Freebase is a huge, multi-domain dataset created by Google, which collects a large amount of entity, attribute, and relationship information. The WN18RR dataset is derived from the WN18 dataset, and the WN18 dataset is derived from the WordNet dataset. Compared with the WordNet dataset, the WN18RR dataset removes the inversion relationship, retains other relationships, and has fewer relationship types. The NELL-995 dataset is an open-source machine learning dataset developed by OpenAI, which is constructed by unsupervised learning and automatically extracts entities and relationships through semantic analysis and pattern recognition of massive web content. YAGO3-10 is a subset of the YAGO dataset, with data from open-source websites or datasets such as Wikipedia, WordNet, and GeoNames.

4.2. Baseline Methods

We use typical representatives of methods based on distributed representation and based on a relational path as the baseline. The former methods comprise TransE, ComplEx, and ConvE, and the latter comprises PRA, DeepPath, and MINERVA.

4.3. Evaluation Protocol

We mainly evaluate the performance of each method through the following two tasks: Relation Link Prediction. This is one of the common tasks to evaluate the reasoning performance of knowledge graphs. Specifically, first, we construct incomplete triples as negative samples, and second, we construct a test dataset containing both positive and negative samples. Then, we employ the Mean Average Precision (MAP) score for the specific evaluation index, which considers both the prediction accuracy and the relative position difference in the sequence.

Entity Link Prediction. This is also one of the common tasks used to evaluate the performance of knowledge graph reasoning. Specifically, we used Hit@N and Mean Reciprocal Rank (MRR). For the value of N in Hit@N, we take N as 1, 3, and 10 to obtain a more comprehensive evaluation result. The specific indicator should be selected in combination with the baseline method.

4.4. Implementation Details

In the experiment, we also need to determine the path length, the number of hidden layers of the neural network, the learning rate, the standard value, and other hyperparameters. We finally choose a path length from 2 to 6, the number of hidden layers from 50 or 100, and the learning rate from 0.01 to 0.05. The standard values are determined based on prior knowledge of each dataset. In addition, the training epochs must also be determined according to different data sets, ranging from 300 to 3000.

4.5. Results

This paper conducts relational link prediction tasks on the NELL-995 and FB15K-237 datasets. Table 2 reports the MAP scores of TransE, PRA, DeepPath, MINERVA, and our model, highlighting that although our model does not perform optimally on every sub-task, it outperforms the baseline model on most tasks, especially on the NELL-995 dataset. Indeed, most individual tasks and the final mean scores were better than the baseline model. Although the highest scores were achieved on only three sub-tasks, the average scores of our model were also very close to the highest average scores on the FB15k-237 dataset.

The entity link prediction task was conducted on the NELL-995, WN18RR, FB15K-237, and YAGO3-10 datasets. Table 3 reports the corresponding results, which reveals that our model performs better on all four datasets compared to the baseline models. Specifically, our model performs best in Hit@1 and Hit@10 and is second only to the ConvE model in Hit@3 and MRR on the NELL-995 dataset. This is because the ConvE model increases the embedding dimension by convolution, making it better at capturing complex information. Our model generally performs well on the FB15k-237 dataset because the FB15k-237 dataset

has the most relationship types and fewer relationship patterns, and the model based on the relational path has an insufficient generalization ability. On the WN18RR and YAGO3-10 datasets, the relational path-based model outperforms the distributed representation-based model, and our model performs best among them. This is because prior knowledge contains more effective information.

Table 2. The MAP scores on NELL995 and FB15K-237.

Dataset	Tasks	TransE	PRA	DeepPath	MINERVA	Our Model
NELL-995	AthletePlaysInLeague	77.3	84.1	92.7	95.2	95.4
	AthletePlaysForTeam	62.7	54.7	75.0	82.4	86.9
	AthleteHomeStadium	71.8	85.9	89.0	89.5	90.7
	TeamPlaySports	76.1	79.1	73.8	84.6	85.5
	OrgHeadquaterCity	62.0	81.1	79.0	94.5	93.6
	BornLocation	71.2	66.8	75.7	79.3	82.2
	OrgHiredPerson	71.9	59.9	74.2	85.1	86.4
	PersonLeadsOrg	75.1	68.1	75.5	83.0	81.4
	...					
	Overall	71.0	72.5	79.4	86.7	87.8
FB15k-237	adjoins	68.4	41.8	69.1	71.8	60.1
	contains	56.7	32.5	39.8	41.5	40.8
	personNationality	44.2	42.1	52.8	62.1	42.9
	filmDirector	41.5	32.8	45.6	38.9	46.3
	filmLanguag	61.5	45.1	52.5	58.9	46.1
	filmWritten	56.1	32.1	36.5	59.1	36.2
	capitalOf	42.5	25.8	43.8	48.9	50.6
	musicianOrigin	38.2	18.5	23.7	23.8	38.7
	...					
	Overall	45.3	31.5	39.8	42.3	45.2

Bold indicates the highest score.

Table 3. Link prediction results (Hits@N and MRR) on datasets WN18RR, FB15k-237, NELL-995 and YAGO3-10.

Dataset	Metric	TransE	Complex	ConvE	MINERVA	Our Model
NELL-995	Hit@1	0.514	0.612	0.672	0.663	0.684
	Hit@3	0.678	0.761	0.808	0.773	0.797
	Hit@10	0.751	0.827	0.864	0.831	0.871
	MRR	0.456	0.694	0.747	0.725	0.740
FB15K-237	Hit@1	0.248	0.303	0.313	0.217	0.228
	Hit@3	0.401	0.434	0.457	0.329	0.464
	Hit@10	0.450	0.572	0.600	0.456	0.491
	MRR	0.361	0.394	0.410	0.293	0.422
WN18RR	Hit@1	0.289	0.382	0.403	0.413	0.440
	Hit@3	0.475	0.433	0.452	0.456	0.485
	Hit@10	0.560	0.480	0.519	0.513	0.542
	MRR	0.359	0.415	0.438	0.448	0.474
YAGO3-10	Hit@1	0.248	0.260	0.350	0.355	0.380
	Hit@3	0.412	0.405	0.490	0.498	0.510
	Hit@10	0.315	0.550	0.620	0.650	0.678
	MRR	0.308	0.360	0.440	0.470	0.492

Bold indicates best performance.

4.6. Analysis

4.6.1. Setting of Standard Values and Ablation Study

We take the experimental results on the NELL-995 dataset as an example to study the influence of different standard values and further measure the influence of prior knowledge

introduced by causal inference. Since the maximum weight of most relationships on the NELL-995 dataset is 0.1, we chose three standard values of 0.01, 0.05, and 0.1 for the following experiment. Furthermore, to ensure the consistency of the experimental results, we still use the Hits@N and MRR indicators to measure the performance differences in this ablation study, where N is still 1, 3, and 10. The experimental results are illustrated in Figure 4.

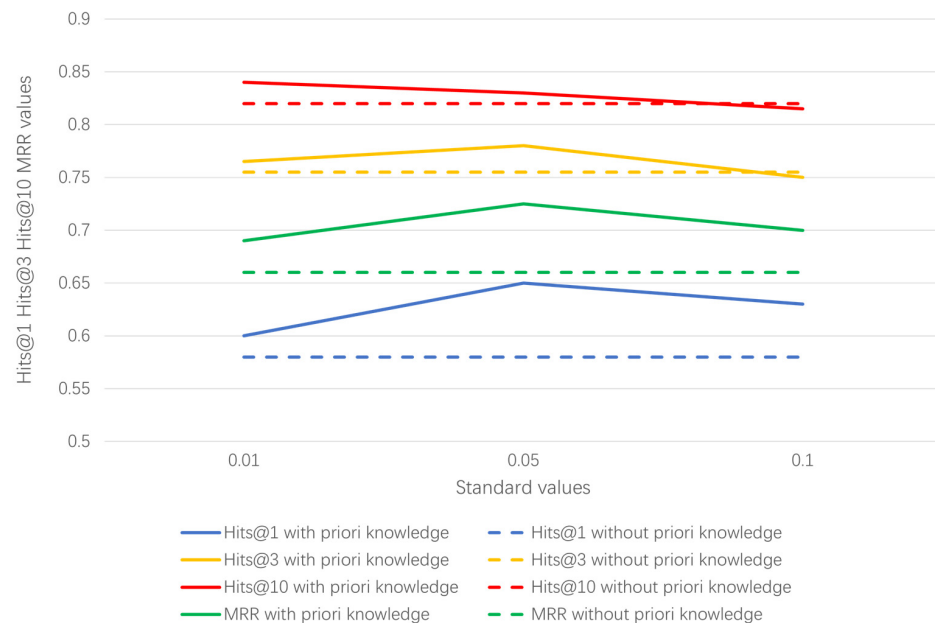


Figure 4. The effect of different standard values on the NELL-995 dataset.

As depicted in Figure 4, after introducing the prior knowledge, the performance of the reinforcement-learning method is significantly improved, demonstrating our method's effectiveness. It should be noted that the method's performance is related to a standard value, as within a certain range, the larger the standard value, the better the method's performance, preserving the standard value between 0.01 and 0.05. Beyond a certain range (for example, the standard value is between 0.05 and 0.10), the method's performance decreases with the standard value's increase. This is because setting the standard value too high may introduce noise. Therefore, it is necessary to set the standard value reasonably, according to different data sets, so that the model obtains the optimal performance.

4.6.2. Impact Analysis of the Number and Length of Paths

In the reasoning process, the agent will face the choice of multiple paths at each step, and the total number of these paths will impact the method's performance. In addition, the length of the relationship path through which the agent reaches the target entity will also affect the method's performance.

Therefore, we first analyze the impact of the total number of optional paths of agents on the method's performance, taking the experimental results on dataset NELL-995 as an example. We first set the length of the agent's relationship path to 4, then set the standard value to 0.1, and finally set the total number of agents' optional paths to 3, 5, 10, and 50. In order to keep the consistency of the experimental results, we still use Hits@N and MRR to evaluate the method's performance, where N is 1, 3, and 10. Figure 5 illustrates the corresponding results, which reveal that the method's performance increases as the number of paths increases within a certain range (the number of paths is less than 5). Beyond a certain range (for example, the number of paths is greater than 10), the number of paths increases, and the method's performance decreases. This may be due to errors in the knowledge graph or inaccurate prior knowledge obtained using the relationship

importance identification module. Hence, the more prior knowledge is introduced, the greater the deviation is.

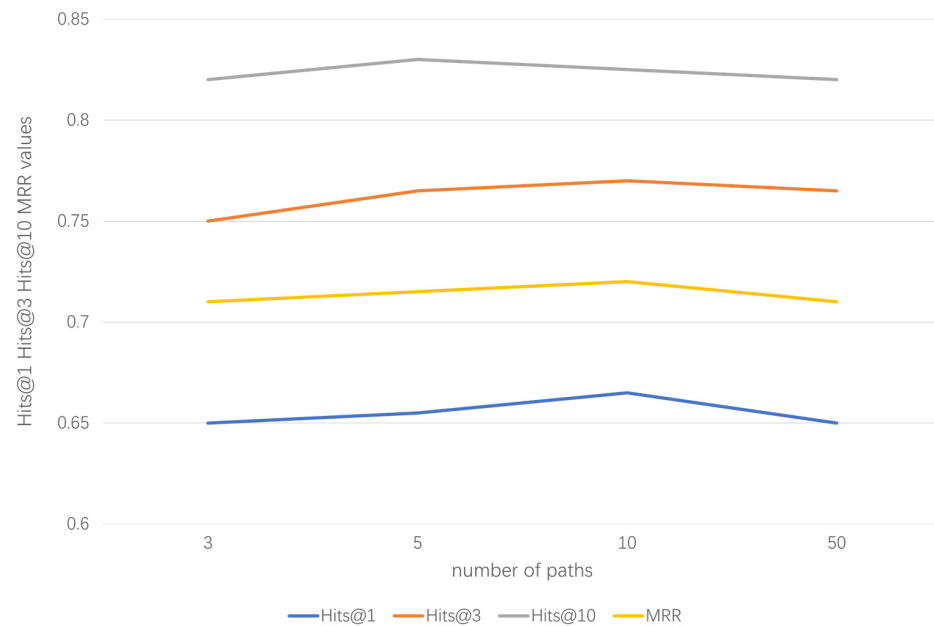


Figure 5. Influence of the number of paths on NELL-995 dataset.

Next, we analyze the impact of path length on the method's performance and conduct experiments on the dataset NELL-995. We fixed the number of paths to 5, set the standard value to 0.1, and set the path length to 2 to 6. The results in Figure 6 infer that within a certain range (path length less than or equal to 4), the method's performance increases with the path length increase. The path length increases beyond a certain range (greater than or equal to 5), but the method's performance is stable. This may be because the increase in path length brings more prior knowledge, which enhances the performance of the method, while the increase in path length also introduces more errors in the knowledge graph itself, which reduces the performance of the method, thus achieving a steady state.

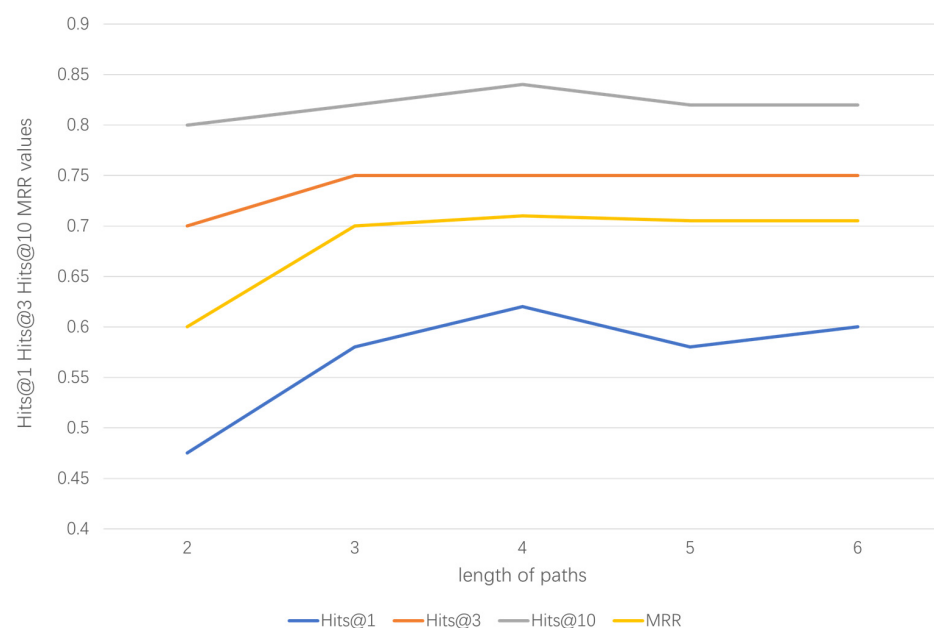


Figure 6. Influence of the length of paths on NELL-995 dataset.

5. Conclusions

This paper proposes a new method of knowledge graph reasoning based on causal reinforcement learning, as existing knowledge graph reasoning methods do not effectively use prior knowledge. Specifically, we use the counterfactual method in causal inference to obtain the weight of the relationship path, obtain the prior knowledge, and introduce it into the reinforcement learning reasoning model. The agent starts from the head entity and then selects the edge through the neural network when all the edge weights are small. Otherwise, it selects the edge according to the prior knowledge. Experiments on four datasets demonstrate the effectiveness of the prior knowledge. Follow-up research will focus on how to make the model automatically learn the standard value and how to effectively combine prior knowledge with semantic information to further improve the performance of reasoning.

Author Contributions: D.L.: conceptualization, writing—original draft, writing—reviewing and editing, methodology, validation, formal analysis. Y.L.: supervision, writing—reviewing and editing. J.W.: supervision, data procession, formal analysis. W.Z.: supervision, writing—editing. G.Z.: supervision, writing—editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset and code generated during the current study are not publicly available because the data and code also form part of the ongoing study, but they can be obtained from the corresponding authors according to reasonable requirements.

Acknowledgments: The authors would like to thank all the reviewers and language editing organizations who helped to improve this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

KGC	Knowledge Graph Completion
SCM	Structural Causal Model
CF-GPS	CounterFactually Guided Policy Search
SARSA	State Action Reward State Action
LSTM	Long Short-Term Memory
MAP	Mean Average Precision
MRR	Mean Reciprocal Rank

References

1. Xiong, C.; Merity, S.; Schoer, R. Dynamic memory networks for visual and textual question answering. In Proceedings of the 33rd International Conference on Machine Learning, Honolulu, HI, USA, 23–26 July 2016; pp. 2397–2406.
2. Logan, R.; Liu, N.F.; Peters, M.E. Using knowledge graphs for fact-aware language modeling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 22–24 August 2019; pp. 5962–5971.
3. Xiong, W.; Yu, S.; Guo, X. Improving question answering over incomplete KBs with knowledge-aware reader. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 22–24 August 2019; pp. 4258–4264.
4. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable AI: A review of machine learning interpretability methods. *Entropy* **2020**, *23*, 180–192. [[CrossRef](#)] [[PubMed](#)]
5. Bordes, A.; Usunier, N.; Garcia-Duran, A. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
6. Wang, Z.; Zhang, J.; Feng, J. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the 28th Association for the Advancement of Artificial Intelligence, San Francisco, CA, USA, 3–6 June 2014; pp. 1112–1119.
7. Lin, X.; Liang, Y.; Giunchiglia, F.; Feng, X.; Guan, R. Compositional learning of relation path embedding for knowledge base completion. *arXiv* **2015**, arXiv:1611.07232.

8. Ji, G.; He, S.; Xu, L. Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 687–696.
9. Pasquale, M.; Pontus, S.; Sebastian, R. Convolutional 2D Knowledge Graph Embeddings. *arXiv* **2018**, arXiv:1707.01476.
10. Trouillon, T.; Welbl, J.; Riedel, S. Complex Embeddings for Simple Link Prediction. *arXiv* **2016**, arXiv:1606.06357.
11. Lao, N.; Mitchell, T.; Cohen, W.W. Rotate: Random walk inference and learning in a large scale knowledge base. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Scotland, UK, 8–11 July 2011; pp. 529–539.
12. Xiong, W.; Hoang, T.; Wang, W.Y. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv* **2017**, arXiv:1707.06690.
13. Das, R.; Dhuliawala, S.; Zaheer, M. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In Proceedings of the ICLR, Vancouver, BC, Canada, 28–30 March 2018; pp. 688–696.
14. Wan, G.; Pan, S.; Gong, C. Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 1926–1932.
15. Wang, H.; Li, S.; Pan, R. Incorporating graph attention mechanism into knowledge graph reasoning based on deep reinforcement learning. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 2623–2631.
16. Peter, J.; Micheal, D.; Nachiketa, C. A framework for causal discovery in non-intervenable systems. *Chaos* **2021**, *31*, 123–142.
17. Eric, J.; Isabel, F.; Ilya, S. Auto-G-Computation of Causal Effects on a Network. *arXiv* **2019**, arXiv:1709.01577.
18. Nicola, G.; Jonas, P.; Sebastian, E. Causal discovery in heavy-tailed models. *arXiv* **2020**, arXiv:1908.05097.
19. Bi, X.T.; Wu, D.; Xie, D. Large-scale chemical process causal discovery from big data with transformer-based deep learning. *Process Saf. Environ. Prot.* **2023**, *173*, 163–177. [[CrossRef](#)]
20. Cui, Y.; Pu, H.; Shi, X. Semiparametric Proximal Causal Inference. *J. Am. Stat. Assoc.* **2023**, *11*, 211–224. [[CrossRef](#)]
21. Wang, Z.; Li, L.; Zeng, D. Incorporating prior knowledge from counterfactuals into knowledge graph reasoning. *Knowl.-Based Syst.* **2021**, *223*, 1307–1323. [[CrossRef](#)]
22. Pearl, J.; Creager, E.; Garg, A. *Causal Inference in Statistics: A Primer*, 1st ed.; John Wiley & Sons: Chichester, NY, USA, 2016; pp. 254–256.
23. Pitis, S.; Pan, S.; Gong, C. Counterfactual data augmentation using locally factored dynamics. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; pp. 2906–2922.
24. Madumal, P.; Miller, T.; Sonenberg, L. Explainable reinforcement learning through a causal lens. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 2493–2500.
25. Lu, C.C.; Huang, B.W.; Schölkopf, B. Sample-efficient reinforcement learning via counterfactual-based data augmentation. *arXiv* **2020**, arXiv:2012.09092.
26. Buesing, L.; Weber, T.; Zwols, Y. Woulda, coulda, shoulda: Counterfactually-guided policy search. In Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 1–5 May 2019; pp. 1693–1710.
27. Moerland, T.M.; Broekens, J.; Plaat, A. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.* **2023**, *16*, 101–118. [[CrossRef](#)]
28. Yi, F.; Fu, W.; Liang, H. Model-based reinforcement learning: A survey. In Proceedings of the 18th International Conference on Electronic Business, Guilin, China, 2–6 December 2018; pp. 421–429.
29. Singh, S.; Jaakkola, T.; Littman, M. Convergence results for single-step on-policy reinforcement-learning algorithms. *Mach. Learn.* **2000**, *38*, 287–308. [[CrossRef](#)]
30. Watkins, C.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [[CrossRef](#)]
31. Mnih, V.; Silver, D.; Graves, A. Playing Atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.09092.
32. Mnih, V.; Silver, D.; Graves, A. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)]
33. Fortunato, M.; Azar, M.; Piot, B. Noisy networks for exploration. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 12–16 August 2018; pp. 1321–1330.
34. Kaelbling, L.P.; Littman, M.; Moore, A. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–255. [[CrossRef](#)]
35. Wang, H.; Huang, T.; Wang, W. Deep reinforcement learning: A survey. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 1726–1744. [[CrossRef](#)]
36. Williams, R. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–246. [[CrossRef](#)]
37. Deng, Z.; Jiang, J.; Long, G.; Zhang, C. Causal Reinforcement Learning: A Survey. *arXiv* **2023**, arXiv:2307.01452.
38. Maximilian, S.; Bernhard, S.; Georg, M. Causal Reinforcement Learning: A Survey. *arXiv* **2016**, arXiv:2106.03443.
39. Guo, J.; Gong, M.; Tao, D. A Relational Intervention Approach for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning. In Proceedings of the 10th International Conference on Learning Representations (Virtual), Toulon, France, 25–29 April 2022; pp. 3453–3470.
40. Pim, D.H.; Dinesh, J.; Sergey, L. Causal Confusion in Imitation Learning. *Statistics* **2022**, *11*, 1467–1480.
41. Deng, Z.; Fu, Z.; Wang, L.; Yang, Z.; Bai, C.; Zhou, T.; Wang, Z.; Jiang, J. False Correlation Reduction for Offline Reinforcement Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *46*, 1199–1211. [[CrossRef](#)] [[PubMed](#)]
42. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]

43. Carlson, A.; Betteridge, B.; Kisiel, B. Toward an architecture for never-ending language learning. In Proceedings of the 25th AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 10–15 June 2010; pp. 1306–1313.
44. Bollacker, K.; Evans, P.; Paritosh, T. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 6–10 May 2008; pp. 1586–1604.
45. Bordes, A.; Weston, R.; Collobert, Y. Learning structured embeddings of knowledge bases. In Proceedings of the 25th AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 16–22 August 2011; pp. 301–306.
46. Sun, Z.; Deng, Z.H.; Nie, J.Y. Rotate: Knowledge graph embedding by relational rotation in complex space. In Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 9–12 May 2019; pp. 1–18.

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