RESEARCH ARTICLE

Knowledge Graph Reasoning with Quantum-Inspired Reinforcement Learning

Junliang Chen[1], **Xianchao Zhang**[2,*], **Fengsong Sun**[1] and **Jun Lu**[2,3,*]

[1] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] Provincial Key Laboratory of Multimodal Perceiving and Intelligent Systems, Jiaxing University, Jiaxing 314001, China
[3] China Academy of Electronics and Information Technology, Beijing 100049, China

Abstract

Knowledge reasoning is a critical task in information fusion systems, and its core step is reasoning missing information from existing facts to improve the knowledge graphs. Embedding-based reasoning methods and path-based reasoning methods are two mainstream knowledge reasoning methods. Embedding-based reasoning methods enable fast and direct reasoning but are limited to simple relationships between entities and exhibit poor performance in reasoning complex logical relationships. Path-based reasoning methods perform better in complex reasoning tasks, but suffer from high computational complexity, a large number of model parameters, and low reasoning efficiency. To address the aforementioned issues, this paper introduces a knowledge reasoning model called Quantum-Inspired Reinforcement Learning (QIRL). QIRL leverages quantum reinforcement learning to train a strategy network via a quantum circuit, aiming to generate and optimize reasoning paths. Quantum circuit achieves complex nonlinear operations through limited quantum reasoning paths gate operations, reducing computational complexity. In addition, this article utilizes the quantum entanglement property to encode high-dimensional data, reducing the number of model training parameters. This article evaluates the QIRL method on entity prediction task and proves that the QIRL method can effectively reduce the number of model training parameters.

Keywords: knowledge graph, knowledge reasoning, reinforcement learning, quantum circuit.

1 Introduction

Knowledge Graphs (KGs) serve as a foundational tool for information fusion, providing a structured representation of knowledge that facilitates data integration and semantic modeling. However, during the construction of KGs, the occurrence of missing facts, resulting from incomplete, inconsistent, or dynamically evolving data, limits their practical applicability. To address this issue, knowledge

Academic Editor:
Xinde Li

Submitted: 25 February 2025
Accepted: 28 April 2025
Published: 25 May 2025

Vol. 2, **No.** 2, 2025.
10.62762/CJIF.2025.552445

***Corresponding authors:**
✉ Xianchao Zhang
zhangxianchao@zjxu.edu.cn
✉ Jun Lu
lujun_cetc@163.com

Citation
Chen, J., Zhang, X., Sun, F., & Lu, J. (2025). Knowledge Graph Reasoning with Quantum-Inspired Reinforcement Learning. *Chinese Journal of Information Fusion*, 2(2), 144–156.

reasoning, as a core task in knowledge graph research, aims to complete missing facts based on existing ones. Embedded-based reasoning method is one of the mainstream [1]. This method captures semantic information [2] and reasoning out missing facts by learning low dimensional vector representations of entities and relationships in KGs and utilizing the relative positions and geometric structures between these embedded vectors. Embedding-based reasoning methods are computationally simple, efficient, and suitable for various applications. However, their representational capacity is limited, making it challenging to reason about complex relationships between entities. For complex reasoning tasks, path-based reasoning methods can effectively capture and model multi-level relationship structures and search for multi-hop paths between entities in the KG, using structural information to reason about missing facts and complete the KG. The path-based reasoning method can effectively handle complex relational reasoning tasks [3], but the path search and matching process usually requires traversing a large number of graph structures, resulting in high computational complexity and significantly increasing the consumption of computing resources. Reducing the number of model parameters can decrease computational complexity, enhance reasoning efficiency, and reduce computational costs.

Traditional methods, while reducing the number of parameters to some extent, still face a trade-off between model performance and computational efficiency. As an emerging computing paradigm, quantum circuit, with their unique quantum advantages, can significantly reduce the number of model training parameters while maintaining or improving model performance, thereby further optimizing computational efficiency and reasoning effectiveness. Arute et al. [7] used a quantum processor called Sycamore to perform an extremely difficult random number generation task and compared it with traditional supercomputers. The experimental results indicate that the time required for Sycamore to complete the task is much shorter than that of classical computers, demonstrating the computational advantage of quantum computers over traditional computers in specific tasks. With the deepening of quantum computing research, variational quantum circuit (VQC) have emerged [8]. This method draws on the principles and structures of artificial neural networks, enabling quantum computing to be effectively applied to machine learning tasks. Chen et al. [10] demonstrated that the parameter space complexity of VQC is $O(N)$, while that of the traditional neural network DON is $O(N^2)$. For KGs, which involve large state and action spaces, VQC can greatly reduce model parameters and computational complexity.

In order to address the challenges of computational complexity and resource consumption in knowledge reasoning models, this paper proposes quantum-inspired reinforcement learning (QIRL), which trains a reinforcement learning strategy network based on quantum circuit and searches for reasoning paths in KGs. Specifically, this paper proposes an embedding-based approach to map the state information of the knowledge graph (KG) into a continuous vector space. This representation is then quantum encoded and input into a strategy network based on quantum circuits for training. Ultimately, the network outputs the probability distribution for the next action. The strategy network based on quantum circuit gradually expands the reasoning path until all reasoning paths in the KG are found. QIRL utilizes the parallelism and entanglement properties of quantum computing, greatly reducing the number of model training parameters, lowering computational complexity, and reducing computational resource consumption. The main contributions of this paper are as follows:

1. This article proposes a knowledge reasoning model based on quantum circuit training reinforcement learning strategy network for the first time, which uses known facts to supervise the training of quantum circuit, update its parameters, and perform knowledge reasoning;
2. This article utilizes the parallelism and entanglement properties of quantum computing to reduce computational complexity, The experimental results show that the QIRL method proposed in this paper can significantly reduce the number of model training parameters.

2 Related Work

2.1 Embedded-Based Reasoning Methods

The embedding-based reasoning methods map entities and relationships in KGs to a low dimensional vector space, allowing their semantic and structural information to be expressed in the form of geometric relationships, thereby efficiently handling reasoning tasks in large-scale KGs. Early methods such as TransE [11], TransH [12] and DistlMult [13] focused on studying geometric relationships such as translation, projection, and linear transformation between embedded vectors of entities and relationships, providing efficient and semantic knowledge representation for knowledge reasoning. Lu et al. [14] proposed the DensE method, which decomposes each relationship into rotation and scaling operators in three-dimensional Euclidean space, which can better represent composite relationships. Pavlovic et al. [15] proposed the ExpressivE method, which uses the spatial relationships of super parallel quadrilaterals to represent the semantic structural information between entities and relationships, providing intuitive geometric explanations for knowledge reasoning tasks. Although embedding based reasoning methods have shown good performance in large-scale KGs reasoning tasks, they are difficult to handle relationships with multi-level semantics or nonlinear structures, and have weak modeling capabilities for complex relationships.

2.2 Path-Based Reasoning Methods

The path-based reasoning methods utilizes the powerful expressive power of neural networks to optimize the reasoning process, which can effectively handle reasoning tasks in large-scale KGs. Dettmers et al. [16] proposed a multi-layer convolutional network mode called ConvE for link prediction, which captures the relationship information between entities through convolutional networks and effectively reasons over large-scale KGs. Yang et al. [17] proposed the hyper relation aware multi-view model HyRel, which learns the globally transferable structure of a graph to reason about unseen graphs, offering high flexibility and usability. Xiong et al. [18] proposed a reinforcement learning framework for learning multi-hop relational paths, which effectively handles complex reasoning tasks with ambiguous answers and scales to large KGs. The path-based reasoning methods perform well in large-scale KG reasoning, but the model requires a large number of training parameters, high computational complexity, and demand significant computing and storage resources.

2.3 Applications of Quantum Computing

The superposition and entanglement properties of quantum computing give it unique advantages in solving certain computational tasks that traditional computers do not have. Chen et al. [10] explored the potential of quantum computing to enhance reinforcement learning and demonstrated the potential of quantum algorithms in improving time and space complexity in simple environments. Chen et al. [4] proposed the asynchronous training of advantage actor-critic variational quantum policies, which reduces the training time of quantum reinforcement learning. As the application of quantum computing in reinforcement learning progresses, quantum reinforcement learning has been extensively explored and applied across various fields. Kim et al. [5] integrated quantum reinforcement learning into reusable rocket control systems, enhancing both computational efficiency and model stability. Ansere et al. [6] proposed a Quantum-empowered Deep Reinforcement Learning (Qe-DRL) approach to enhance computational learning speed and task processing efficiency for IoT devices under quantum uncertainty and time-varying channel conditions. In addition, quantum reinforcement learning is widely used in fields such as biochemistry and medicine, and has shown good performance. However, the above application scenarios are generally relatively simple, usually requiring only a limited number of qubits and small state and action spaces. KGs typically contain large states and action spaces. To reduce the computational complexity of reasoning models in large-scale KGs, this paper proposes a quantum reinforcement learning based knowledge reasoning model QIRL, which greatly reduces the training parameters of the model by utilizing quantum advantages. The performance of the proposed model is validated on universal datasets. In contrast to emerging LLM-based reasoning methods [9], our approach emphasizes computational efficiency and parameter reduction via quantum-inspired techniques.

3 Problem Description

A KG contains a large amount of entity and relational data. Given a graph $G = \{E, \epsilon, R\}$, E represents the set of entities and R represents the set of relationships between entities. $\epsilon = \{(h_i, r_i, t_i) | h_i, t_i \in E, r_i \in R\}$ is the set of triplets in a KG, each triplet consisting of a head entity h_i, a relationship r_i, and a tail entity t_i. Due to incomplete data sources and insufficient information extraction, KGs often have missing elements, which can be filled in through knowledge reasoning. According to the type of missing elements, knowledge reasoning tasks can be divided into head reasoning $(?, r_i, t_i)$, tail reasoning $(h_i, r_i, ?)$ and relational reasoning $(h_i, ?, t_i)$, among which $?$ indicates missing elements. The path-based knowledge reasoning methods reason about missing facts by searching for paths between entities. As the number of entities and relationships in the KG increases, the number of possible reasoning paths increases exponentially, which makes a comprehensive analysis of these paths computationally complex. Therefore, how to reduce computational complexity while ensuring reasoning accuracy has become a key challenge in current research.

To address the above issues, this paper proposes a quantum reinforcement learning-based knowledge reasoning method, QIRL, which utilizes quantum circuits to train the strategy network and reduce the computational complexity of the training process. The QIRL method consists of two main components: the external environment and the quantum strategy network. The quantum strategy network is primarily composed of quantum circuits, which in turn consist of quantum bits, quantum gates, and measurement operations [5]. Quantum bits are the fundamental units used to construct quantum circuits, while quantum gates serve as the basic units for manipulating quantum bits. Quantum gates perform linear transformations on the states of quantum bits, create entanglement between them, and enable parallelism in quantum computation. This allows quantum circuits to simultaneously process data from multiple quantum bits, significantly reducing the number of model training parameters and lowering computational complexity. After the quantum circuit processes the data of n quantum bits in parallel, a measurement is performed on these n quantum bits to obtain the output of the quantum strategy network. The quantum strategy network generates the next action based on the current external environment state and continuously updates its parameters until the reasoning process is completed. This paper mainly focuses on tail reasoning tasks, while the other two types of reasoning tasks can be similarly transformed into the form of tail reasoning tasks.

4 QIRL Model

In this section, an overview of the proposed QIRL method is first introduced. The QIRL method trains a reinforcement learning strategy network based on quantum circuit and continuously updates the quantum circuit parameters according to the state of the agent. Afterwards, the training and reasoning process of the QIRL method were described in detail.

4.1 Modeling

The QIRL method updates the parameters of the quantum policy network by continuously interacting with the environment to find the optimal reasoning paths. In this paper, the environment is modeled as a Markov Decision Process (MDP) and the interaction dynamics between the agent and the KG are specified. MDP can be represented by tuple $\langle S, R, P, R' \rangle$, where $S = \{s_1, s_2, \cdots, s_n\}$ represents the continuous state space of the KG as input to the quantum policy network. $R = \{r_1, r_2, \cdots, r_n\}$ represents the set of relationships in the KG, that is, the set of all next optional actions output by the quantum policy network. $P\left(S_{i+1} = s' | S_i = s, R_i = r\right)$ is a state transition probability function that represents the probability of the QIRL policy network transitioning from state s to state s' upon selecting action r. $R'\left(s, r\right)$ is the reward function for each pair (s, r), designed to encourage the agent to learn the optimal strategy. The specific settings of the QIRL method are as follows:

- Actions: The actions of the quantum policy network can be represented by the relationship r in the KG. For the entity pair (h_i, t_i), starting from the head entity h_i, the agent selects the action with the highest probability as the next action r_i based on the output action probability distribution of the quantum policy network, to extend the path until reaching the real tail entity t_i; To ensure the consistency of the output dimension of the strategy network, this paper defines the action space as all relationships and their inverse relationships in the KG, and the dimension of the action space is consistent with the number of quantum bits in the quantum circuit.
- States: KGs contain a large number of entities and relationships, which are discrete symbols, while policy network training needs to be conducted in a continuous space. In order for the policy network to better capture semantic information, it is necessary to transform entities and relationships into continuous vectors. This model employs the translation-based embedding method TransE, which maps the discrete symbols of entities and relationships into a continuous vector space, capturing the agent's position in the KG and enabling transitions from the current entity to the next via the selected relationship. The state of the agent in step i can be represented by the following vector:
$$s_i = (e_i, e_t - e_i) \tag{1}$$
where e_i represents the embedding vector of the current entity, e_t represents the embedding vector of the tail entity, and $e_t - e_i$ represents the distance between the target entity and the current entity in the vector space. After the embedding vectors of entities and relationships are trained,

they still need to undergo quantum encoding before being input into the quantum circuit of the strategy function for processing. Quantum amplitude coding is an efficient mapping method from classical data to quantum states where n qubits can encode states with a dimension of 2^n, achieving compressed representation of high-dimensional input data. This paper uses quantum amplitude coding to encode states s_i. The state s_i is normalized to $\bar{s}_i$, where $\|s_i\|$ denotes the norm of s_i.

$$\bar{s}_i = \frac{s_i}{\|s_i\|} \tag{2}$$

assuming that the dimension of s_i is 2^n, it can be encoded into an n-qubit quantum state expressed as:

$$|\varphi\rangle = \bar{s}_{i,0} \cdot |00\cdots 0\rangle + \bar{s}_{i,1} \cdot |00\cdots 1\rangle + \cdots + \bar{s}_{i,2^n-1} \cdot |11\cdots 1\rangle \tag{3}$$

where $|00\cdots 0\rangle, |00\cdots 1\rangle, \cdots$ represent the possible states of the n-qubit system.

- Rewards: Select actions one by one from the head entity until reaching the real tail entity, and this action sequence is the reasoning path of the quantum policy network. Due to the large action space available to agents, there are far more incorrect action decision sequences than correct ones, and the number of incorrect decision sequences increases exponentially with the length of the reasoning path. To encourage agents to find the correct reasoning path, this paper defines the reward function of the policy network as:

$$R'_{total} = r'_{GLOBAL} + r'_{EFFICIENCY} + r'_{DIVERSITY} \tag{4}$$

where r'_{GLOBAL} represents the overall reward, which can be expressed by the following formula:

$$r_{GLOBAL'} = \begin{cases} +1, if the path reaches t_i \\ -1, otherwise \end{cases} \tag{5}$$

if the agent reaches the real tail entity after selecting a series of actions, it will receive a positive reward +1; otherwise, it will receive a negative reward -1. The length of the reasoning path also has a significant impact on the reasoning results. As the path length increases, the accuracy and reliability of the information decrease, accompanied by cumulative errors. Therefore, shorter reasoning paths are generally more reliable than longer ones. This paper reduces path length and improves reasoning efficiency by limiting the interaction length between the agent and the environment. The definition of the efficiency reward function is as follows:

$$r_{EFFICIENCY'} = \frac{1}{length\,(p)} \tag{6}$$

where $P : r_1 r_2 \cdots r_n$ represents the reasoning paths. The agent search for reasoning paths by learning normal samples, which have similar representations in vector space. This makes the agent more inclined to search for reasoning paths with similar semantics, which often contain redundant information. To enable the agent to find different paths as much as possible, this paper defines the path diversity reward function as follows:

$$r_{DIVERSITY'} = -\frac{1}{|F|}\sum_{i=1}^{|F|} \cos\left(p, p_i\right) \tag{7}$$

where $p = \sum_{i=1}^{l} r_i$ represents the path embedding of the reasoning path P, and F represents the total number of reasoning paths. The strategy network updates the parameters of quantum circuit by constantly interacting with the environment.

- Policy network: This paper trains a quantum policy function $\pi\left(r\,|\xi\left(s_i\right); w\right) = p\left(r|s_i; w\right)$ based on the quantum circuit, where w represents the parameters of the quantum circuit, $\xi\left(s_i\right)$ represents the quantum amplitude encoding of the state s_i, and $\pi\left(r\,|\xi\left(s_i\right); w\right)$ represents the probability of selecting the action r in state s_i when the parameters of the quantum circuit are w. The quantum strategy network constantly interacts with the external environment, updating the parameters of the quantum circuit under the constraint of the reward function to find the optimal reasoning paths.

When training the policy network using the QIRL method, the current environmental state s is first input into the quantum policy network and quantum encoded to initialize the input state of the quantum circuit. After quantum computing, measure the probability distribution of actions, select the action with the highest probability as the next action r, and update the environmental state to s'. The reward function calculates the reward value R' based on the current environmental state and inputs it together with s' into the optimizer of the quantum strategy

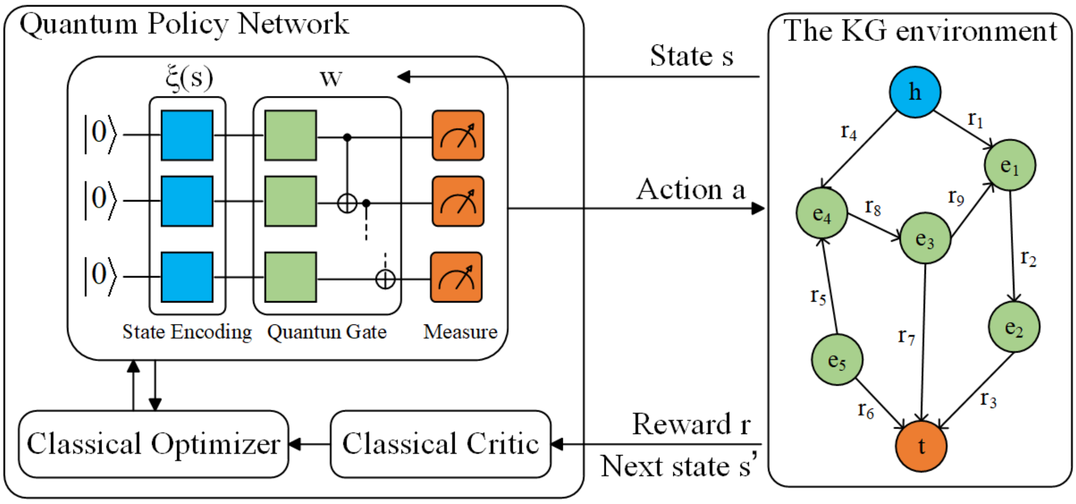


Figure 1. Mimic octopus.

network to update the quantum circuit parameters w. After training, the quantum strategy network generates reasoning paths as logical rules for entity reasoning to reason about missing entities. The flow of the QIRL method is illustrated in Figure 1.

4.2 Strategy Network Training

For each entity pair (h_i, t_i), a random breadth first search (BFS) is used to find all paths between the entity pairs. This paper introduces a random mechanism in the BFS algorithm, which randomly selects an intermediate node instead of directly searching for paths between the head entity h_i and tail entity t_i. Then perform two BFS between $(h_i, inter_{\mathrm{i}})$ and $(inter_{\mathrm{i}}, t_i)$ to search for the correct paths faster and improve the convergence speed of the model. For entity pairs (h, t), randomly select the inter mediate entity $inter$ and perform BFS between $(h, inter)$ and $(inter, t)$ to find paths. Entities are usually associated with multiple different relationships. Assuming that the head entity h connected to the new entity e_1 through the relationship r_1, and then e_1 connected to the entity $inter$ through the relationship r_2, a reasoning path $r_1 \to r_2$ is obtained. $inter$ connected to the new entity e_2 through relationship r_3, and e_2 is connected to the tail entity t through relationship r_4. Therefore, the reasoning path is $r_3 \to r_4$. Combining the two reasoning paths together, the final reasoning path between (h, t) is $P = r_1 \to r_2 \to r_3 \to r_4$.

This paper conducts supervised training on the quantum policy network of QIRL method based on the paths between entity pairs searched by random BFS method. The quantum strategy network maximizes the expected cumulative reward by continuously updating the parameters of the quantum circuit w. The expected cumulative reward $J(w)$ is expressed as follows:

$$J(w) = \sum_i \sum_{r \in R} \pi\left(r \,|\xi\left(s_i\right); w\right) R'_{s_i, r_i} \tag{8}$$

where R'_{s_i, r_i} represents the reward value obtained by selecting action r_i in state s_i. For each supervised path, the agent receives a reward of +1 for each successful search and updates the quantum circuit parameters using the approximate gradient of the Monte Carlo strategy gradient:

$$\begin{aligned} \nabla_w J(w) &= \sum_i \sum_{r \in A} \pi\left(r \,|\xi\left(s_i\right); w\right) \nabla_w \log \pi\left(r \,|\xi\left(s_i\right); w\right) \\ &= \nabla_w \sum_i \log \pi\left(r = r_i \,|\xi\left(s_i\right); w\right) \end{aligned} \tag{9}$$

where r_i belongs to path P, which represents the action with the highest probability of the quantum circuit output in step i.

In the pretraining process of the quantum strategy network mentioned above, the reasoning paths obtained often contain a large amount of redundant information and similar paths. To improve reasoning efficiency, this paper retrains the quantum policy network using a reward function to find a more efficient reasoning path controlled by the reward function. For each entity pair (h_i, t_i), the agent selects an action from the head entity h_i based on the output value of the quantum policy function $\pi\left(r \,|\xi\left(s_i\right); w\right)$ to expand the reasoning paths. If the selected action cannot connect to any entity, the agent receives a negative reward and remains in its original state. Since agents follow the quantum policy function to extend the reasoning paths, they will not get stuck due to repeating incorrect steps. This paper improves training efficiency by limiting the maximum length of the reasoning paths. If the agent finds the correct tail entity t_i within the maximum path length, a new reasoning path is generated. Conversely, if the agent fails to locate the correct tail entity t_i within the constrained path length, the current training iteration is terminated. The gradient function of the quantum strategy network during the retraining process is as follows:

$$\nabla_w J(w) = \nabla_w \sum_i \log \pi \left(r = r_i \left| \xi \left(s_i \right) ; w \right. \right) R'_{total} \tag{10}$$

The retraining process of the model is shown in Algorithm 1, where w is the parameters of the quantum circuit, $\xi(s_t)$ is the quantum amplitude encoding of the input state s_t, and M_neg represents the set of negative steps.

Algorithm 1: Retraining the policy function

```
Data: $w$
for episode < $N$ do
Initialize state vector $s_i \leftarrow s_0$
Initialize episode length $steps \leftarrow 0$
encode $s_i$ with Amplitude Encoding $\xi(s_i)$
while $steps < max\ length$ do
    Probability distribution of action
     $r \sim \pi(r|\xi(s_i))$;
    Observe reward $R_i'$, next state $\xi(s_(i+1))$;
    if $R_i' = -1$ then
        Save $< \xi(s_i), r >$ to $M_neg$
    end
    if success or steps=max length then
        then Break
    end
end
Increase steps
Update $w$ using
    $g \propto \nabla_w \sum_{M_{neg}} \log \pi \left( r = r_i \left| \xi \left( s_i \right) ; w \right. \right) (-1)$
If success then
    $R_total \leftarrow \lambda_1 r_{GLOBAL} + \lambda_2 r_{EFFICIENCY} + \lambda_3 r_{DIVERSITY}$
Update $w$ using
    $g \propto \nabla_w \sum_i \log \pi \left( r = r_i \left| \xi \left( s_i \right) ; w \right. \right) R'_{total}$
If reach $t_i$ then
reasoning Path $P' \leftarrow r_1 r_2 \cdots r_i$
```

4.3 Entity Reasoning

Entity reasoning refers to reasoning about missing entity information from known entities and their relationships through the relationships or attributes between entities in a KG. Due to the large number of complex relationships between entities in KGs, conducting path searches one by one will result in high time costs. The bidirectional path constrained search algorithm [18] is an efficient path search algorithm that can simultaneously search for paths from both positive and negative directions, significantly reducing the number of search paths. Therefore, this paper adopts a bidirectional path constrained search algorithm to search for paths between entity pairs, significantly reducing the search space and improving the efficiency of path search.

For an entity pair (h, t), the reasoning path P' trained by the QIRL method is used as a logical formula, starting from the head entity h and tail entity t respectively, and gradually expanding the reasoning paths through a bidirectional path constraint search algorithm. If the intersection of the path entities in both directions of the bidirectional path constraint search algorithm is not empty, the true tail entity can be successfully found based on the reasoning path P'. Otherwise, the tail entity cannot be successfully reasoned.

Assuming there is a reasoning path $P = r_1 \rightarrow r_2$, perform bidirectional path constraint search on entity pair (h, t). Starting from the head entity h and connecting to the entity e through relationship r_1, the forward entity collection is $left = \{e\}$. At this point, if the length of the forward entity set $left$ is 1 and the length of the reverse entity set $right$ is 0, the next step is to expand $right$. Starting from the tail entity t and connecting to the entity e through relationship r_2, $right = \{e\}$, at this point, the intersection of $left$ and $right$ is not empty, the tail entity is successfully reasoned out.

5 Comparison and Analysis of Model Parameters

Quantum circuit is mainly composed of three parts: quantum bits, quantum gates, and measurements. Quantum bits exhibit unique quantum advantages compared to classical bits due to their properties of superposition states. Quantum amplitude coding is an efficient method for mapping classical data to quantum states. It encodes states with 2^{N_q} dimensions into amplitude values of N_q quantum bits, achieving compressed representation of high-dimensional input data. The number of quantum bits in a quantum circuit depends on the dimensions of the state and action space of the KG. Assuming that the state space dimension is $\dim_s$ and the action space dimension is $\dim_a$, the number of quantum bits N_q needs to satisfy:

$$\begin{aligned} 2^{N_q} &\geq \dim_s \\ N_q &\geq \dim_a \end{aligned} \tag{11}$$

Quantum gate is the fundamental unit that operates on qubits, achieving entangled states between qubits by linearly transforming their states, and establishing complex connections between multiple qubits. CNOT gates are commonly used to generate entangled states between quantum bits, achieving parallelism in quantum computing. After passing through the CNOT gate, the quantum bits of the input circuit are rotated through the Rot gate to achieve precise control over the state of the quantum bits. The x, y, z Rot gate in quantum circuit manipulates a three-dimensional weight vector (W_x, W_y, W_z) to achieve rotation of quantum bits in three dimensions. Therefore, the parameter number of each layer of the Rot gate in quantum circuit is:

$$3 \times N_q \tag{12}$$

if the number of layers in a quantum circuit is L_q, the parameter number of the entire quantum circuit's Rot gate, denoted as N_R,is:

$$N_R = 3 \times N_q \times L_q \tag{13}$$

for each quantum circuit, if a bias vector is set as B, the parameter number of the bias term N_B is:

$$N_B = N_q \tag{14}$$

the parameter number of the entire quantum circuit N, can be expressed as:

$$N = N_R + N_B = 3 \times N_q \times L_q + N_q = (3L_q + 1)\, N_q \tag{15}$$

DeepPath [18] is trained on a reinforcement learning network, which consists of two hidden layers and one output layer. Assuming that the state space dimension $\dim_s = 200$, the action space dimension $\dim_a = 16$, and the dimension of the first hidden layer is 512, the total number of parameters from input to the first hidden layer N_{D_1} is:

$$N_{D_1} = 200 \times 512 + 512 = 102912 \tag{16}$$

the dimension of the second hidden layer is 1024, so the total number of parameters from the first hidden layer to the second hidden layer N_2 is:

$$N_{D_2} = 512 \times 1024 + 1024 = 525312 \tag{17}$$

the total number of parameters from the second hidden layer to the output layer N_3 is:

$$N_{D_3} = 1024 \times 16 + 16 = 16400 \tag{18}$$

the total number of parameters for the DeepPath network N_D is:

$$N_D = N_{D_1} + N_{D_2} + N_{D_3} = 1271824 \tag{19}$$

The NCRL [24] is trained based on a neural network that includes an Embedding layer, an LSTM layer, a Linear layer, and an Attention layer. Assuming the action space dimension is 16 and the Embedding layer dimension is 1024, the number of parameters in the Embedding layer N_{N_1} is:

$$N_{N_1} = (16 + 1) \times 1024 = 17408 \tag{20}$$

the parameters of the LSTM layer include the weights and biases for the input-to-hidden and hidden-to-hidden connections. Assuming both the input and hidden layer dimensions are 1024, the number of parameters in the LSTM layer N_{N_2} is:

$$\begin{aligned} N_{N_2} &= 4 \times (1024 \times 1024 + 1024 + 1024 \times 1024 + 1024) \\ &= 8396800 \end{aligned} \tag{21}$$

assuming the input and output dimensions of the Linear layer are both 1024, the number of parameters N_{N_3} can be expressed as:

$$N_{N_3} = 1024 \times 1 + 1 = 1025 \tag{22}$$

the Attention layer consists of three Linear layers, so the number of parameters in the Attention layer N_{N_4} can be expressed as:

$$N_{N_4} = 3 \times (1024 \times 1024 + 1024) = 3148800 \tag{23}$$

therefore, the total number of parameters in the NCRL model N_N is:

$$N_N = 17408 + 8396800 + 1025 + 3148800 = 11564033 \tag{24}$$

Under the same parameter settings, the number of parameters required for quantum circuit is $N = (3 \times 3 + 1) \times 16 = 160$, significantly reducing the number of parameters needed for model training.

Quantum circuits, through quantum superposition and entanglement, can represent high-dimensional states with a small number of qubits. However, this compression capability has not been fully realized in current classical-based quantum simulators, where the storage and computational demands far exceed those required for actual operation on quantum hardware [27]. Classical simulators need to explicitly store the entire quantum state vector, i.e., all amplitudes [26], whereas in actual quantum hardware, these values are implicitly encoded in the physical system and do not require explicit storage. Furthermore, classical simulators cannot achieve exponential parallelism via quantum superposition states as quantum hardware can, and thus must simulate all possible paths sequentially, increasing the processing time requirement.

6 Experiments

In this section, we first introduce the experimental setup, including the dataset, baseline, evaluation metrics, and parameter settings. Then, the main results of the proposed model were introduced, and all baselines were compared on two benchmark datasets. Furthermore, the training parameter quantities of different models were analyzed and compared.

6.1 Database and Settings

To evaluate the performance of the QIRL method in entity prediction tasks, partial KG data was extracted from two benchmark datasets, Kinship and YAGO3-10, for testing. The Kinship dataset is small but logically complex, testing the model's reasoning ability with sparse data. The YAGO3-10 dataset is large and diverse, evaluating the model's scalability and generalization.The increase in the number of relationships in knowledge reasoning enhances the completeness of reasoning paths and improves the accuracy of knowledge reasoning. Constrained by the computational hardware's processing capabilities, the quantum reinforcement learning model in this paper contains 16 qubits and can handle 16 relationships in the dataset. To find the reasoning path more efficiently, breadth first search is performed from both the head entity and tail entity directions. The action space includes the relationships between entities and their inverse relationships. The statistical data of the extracted dataset is shown in Table 1:

Table 1. Statistical data of the dataset.

Dataset	entities	rel	train	Test	valid
Kinship	104	16	4897	1226	581
YAGO3-10	123183	16	313569	3867	3572

The QIRL model parameters proposed in this paper include the quantum bit count num_qubits of the quantum circuit, the learning rate lr of the model training, the entity embedding dimension $embedding_dim$, the reinforcement learning discount rate γ, the maximum number of attempted steps for reasoning path search max_length, etc., where num_qubits is the number of relationships contained in the dataset. To improve the generalization ability and accuracy of the model, this paper generates positive and negative samples of the test set data based on pra [25]. This paper uses a TransE based method to train embedding vector representations of entities and relationships. The optimal parameters for the model are $embedding_dim = 100$, $\gamma = 0.99$ and $max_length = 50$.

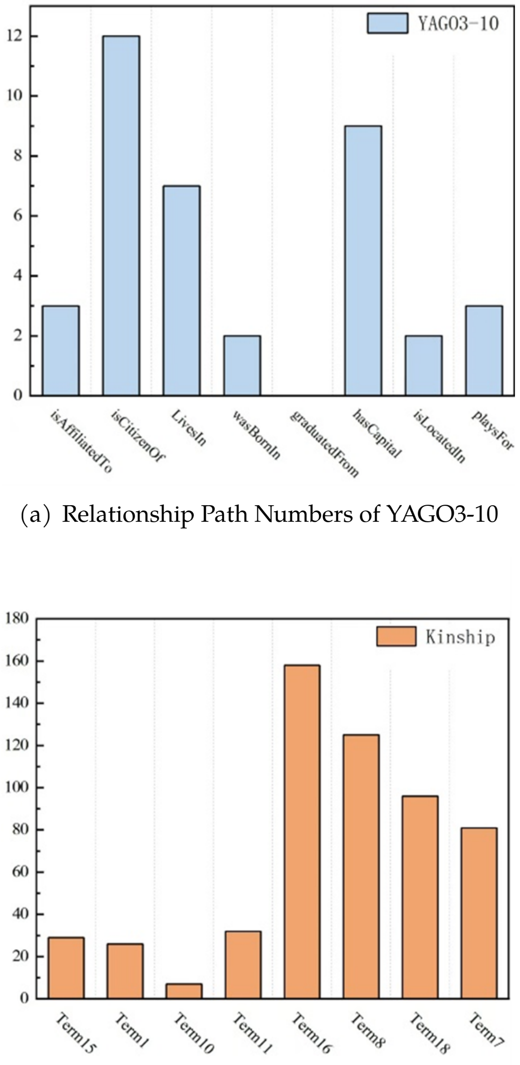
6.2 Baseline and Evaluation Index

This paper compares embedding-based methods and path-based methods to explore their performance on the dataset used in this paper. Embedding-based methods include translation-based method TransE [11], rotation-based method RotalE [19] and complex vector space-based method CompleEx [20]. Path-based methods include neural network-based and logical reasoning combined method NeuralLP [21], deep learning-based and relational reasoning combined method DRUM [22], rule-based method RulE [23], neural network-based and symbolic reasoning combined method NCRL [24], and reinforcement learning-based method DeepPath [18]. In order to ensure the fairness of the experiment., this paper used the publicly released source code of each model and adopted the best hyperparameters provided in the original text for the experiment.

Table 2. Comparison of relationship path numbers.

Dataset	Average Path number
Kinship	69
YAGO3-10	5

In this experiment, Mean Average Precision (MAP) and Accuracy were used as evaluation metrics. This



(a) Relationship Path Numbers of YAGO3-10

(b) Relationship Path Numbers of Kinship

Figure 2. Comparison of relationship path numbers.

paper primarily focuses on the tail reasoning task and evaluates it on the tail entity reasoning task, namely predicting the entity represented by ? in $(h_i, r_i, ?)$. For each test triplet $(h_i, r_i, ?)$, start from the head entity and search for the tail entity based on the generated reasoning paths. Tail reasoning tasks are the main focus of this paper, while head reasoning tasks and relationship reasoning tasks can also be converted into this paradigm.

6.3 Main Results

Figure 2 shows the entity reasoning MAP values of the QIRL method on different relationships between the Kinship and YAGO3-10 datasets, and Table 2 lists the number of reasoning paths found by the QIRL method on the Kinship and YAGO3-10 datasets. The agent found more reasoning paths in the Kinship dataset than in the YAGO3-10 dataset, which means the agent can find strongly correlated reasoning paths in the YAGO3-10 dataset and filter out similar or unrelated ones, but it is difficult to find the most relevant reasoning paths in the Kinship dataset. In addition, when the number of reasoning paths is too small, it is also difficult to find sufficient reasoning evidence to obtain reasoning results.

As shown in Table 3, the test results of QIRL method on YAGO3-10 are better than those on Kinship, because the correlation between entities in Kinship is low, and QIRL cannot find sufficient reasoning evidence on Kinship. However, in YAGO3-10, the correlation between entities is high and the QIRL method can find sufficient reasoning evidence for reasoning, thus obtaining good test results.

Table 3. Reasoning results.

Model	Kinship		YAGO3-10	
	MAP	Accuracy	MAP	Accuracy
TransE[11]	0.325	0.237	0.072	0.099
RotalE[19]	0.875	0.744	0.071	0.082
ComplEx[20]	0.806	0.651	0.041	0.051
NCRL[24]	0.366	0.337	0.950	0.950
NeuralLP[21]	-	0.088	-	0.001
DRUM[22]	-	0.135	-	0.004
RulE[23]	0.665	0.649	0.713	0.697
DeepPath[18]	0.307	0.314	0.658	0.594
QIRL	0.326	0.312	0.661	0.594

Table 4. Comparison of parameter numbers in different models.

Model	Parameter number
QIRL	160
DeepPath	644624
NeuralLP	139936
DRUM	204513
RulE	4722205
NCRL	11564033

Compared with embedding-based methods, path-based methods such as QIRL perform worse on Kinship. Embedding-based methods are reasoning through distance or similarity measures. The relationships in Kinship are mostly simple and symmetrical, and most relationships can be directly learned through geometric relationships in the embedding space. However, path-based methods such as QIRL do not fully utilize the simple structures

Table 5. Comparison of model performance on accuracy and parameter number.

Model	Kinship		YAGO3-10		Parameter	Parameter
	MAP	Accuracy	MAP	Accuracy	number	number Improvement
DeepPath[18]	0.314	-	0.594	-	644624	-
NeuralLP[21]	0.088	-71.975%	0.001	-99.832%	139936	+78.292%
DRUM[22]	0.135	-57.006%	0.004	-99.327%	204513	+68.274%
NCRL[24]	0.337	7.325%	0.950	59.33%	11564033	-1693.919%
RulE[23]	0.649	106.688%	0.697	17.340%	4722205	-632.552%
QIRL	0.326	-0.637%	0.594	-	160	+99.975%

in the data, so embedding-based models can better leverage their advantages. On YAGO3-10, the performance of embedding-based methods significantly decreases due to the complexity of YAGO3-10 relationship reasoning tasks. At this point, path-based methods such as QIRL can better capture the complex relationships between entities.

The performance of QIRL on Kinship and YAGO3-10 is inferior to path-based methods such as RulE and NCRL. RulE learns explicit logical rules for reasoning and directly utilizes the inherent structure and relationships in the data. Therefore, it can achieve better results on Kinship dataset with clear regularity in relationships. NCRL combines neural networks and logical reasoning, using graph neural networks to process graph structured data, which can effectively learn potential patterns between entities and relationships. Therefore, it can achieve good results in the YAGO3-10 dataset with complex relationships and no clear rules.

Table 4 shows the number of network training parameters for different models. Through comparative analysis, it can be clearly seen that compared to other methods, the QIRL algorithm requires significantly fewer parameters in the model training process, only requiring hundreds of parameters to effectively train the model and achieve excellent results. This feature enables the QIRL algorithm to significantly reduce the computational resource consumption of the model while maintaining efficient training. By significantly reducing the number of parameters, QIRL not only demonstrates significant advantages in memory and computing resources, but also effectively avoids problems such as overfitting, thereby improving the model's generalization ability and robustness in practical applications. Overall, the QIRL algorithm has improved model performance and significantly enhanced computational efficiency by optimizing the number of parameters, demonstrating its potential in complex tasks.

As shown in Table 5, QIRL significantly reduces the number of model parameter while maintaining high model accuracy. In contrast, other methods typically increase the training parameters to improve accuracy or sacrifice model accuracy in order to reduce the number of parameter. Table 5 compares exclusively neural network-based methods to ensure parameter computability.

7 Conclusion

This paper proposes a quantum reinforcement learning-based knowledge reasoning method, QIRL, which significantly reduces the number of model's training parameter and computational complexity by leveraging quantum advantages through the construction of a quantum circuit to train the policy network. However, most quantum computing still relies on classical computer simulators to test and validate quantum algorithms, and these simulators have much higher storage and computational demands compared to the actual requirements when running on quantum hardware. In the future, further optimization of the quantum reinforcement learning model's performance is expected, aiming to reduce the number of model's training parameter while improving reasoning accuracy.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported by the Project of Natural Science Foundation of Zhejiang Province under Grant LD24F020009.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Chen, Y., Li, H., Li, H., Liu, W., Wu, Y., Huang, Q., & Wan, S. (2022). An overview of knowledge graph reasoning: key technologies and applications. *Journal of Sensor and Actuator Networks*, *11*(4), 78. [CrossRef]

[2] Chen, S., Yang, X., & Li, Z. (2023). Improving semantic segmentation with knowledge reasoning network. *Journal of Visual Communication and Image Representation*, *96*, 103923. [CrossRef]

[3] Zhang, Y., & Yao, Q. (2022). Knowledge graph reasoning with relational digraph. *Proceedings of the ACM Web Conference 2022*, 912–924. [CrossRef]

[4] Chen, S. Y. C. (2023). Asynchronous training of quantum reinforcement learning. *Procedia Computer Science*, *222*, 321–330. [CrossRef]

[5] Kim, G. S., Chung, J., & Park, S. (2024). Realizing Stabilized Landing for Computation-Limited Reusable Rockets: A Quantum Reinforcement Learning Approach. *IEEE Transactions on Vehicular Technology*, *73*(8), 12252–12257. [CrossRef]

[6] Ansere, J. A., Gyamfi, E., Sharma, V., Shin, H., Dobre, O. A., & Duong, T. Q. (2024). Quantum Deep Reinforcement Learning for Dynamic Resource Allocation in Mobile Edge Computing-Based IoT Systems. *IEEE Transactions on Wireless Communications*, *23*(6), 6221–6233. [CrossRef]

[7] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., ... & Boixo, S. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, *574*(7779), 505–510. [CrossRef]

[8] Kruse, G., Dragan, T., Wille, R., & Lorenz, J. M. (2023). Variational Quantum Circuit Design for Quantum Reinforcement Learning on Continuous Environments. *International Conference on Agents and Artificial Intelligence*. [CrossRef]

[9] Guo, T., Yang, Q., Wang, C., Liu, Y., Li, P., Tang, J., Li, D., & Wen, Y. (2024). Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph. *Complex & Intelligent Systems*, *10*(5), 7063–7076. [CrossRef]

[10] Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007–141024. [CrossRef]

[11] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*, *26*.

[12] Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, *28*(1). [CrossRef]

[13] Yang, B., Yih, W. T., He, X., Gao, J., & Deng, L. (2014). Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *International Conference on Learning Representations*. [CrossRef]

[14] Lu, H., Hu, H., & Lin, X. (2022). DensE: An enhanced non-commutative representation for knowledge graph embedding with adaptive semantic hierarchy. *Neurocomputing*, *476*, 115–125. [CrossRef]

[15] Pavlović, A., & Sallinger, E. (2023). ExpressivE: A Spatio-Functional Embedding For Knowledge Graph Completion. *The Eleventh International Conference on Learning Representations*. [CrossRef]

[16] Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). [CrossRef]

[17] Yang, J., Jiang, X., Gao, Y., Yang, L. T., & Yang, J. (2024). Generalize to Fully Unseen Graphs: Learn Transferable Hyper-Relation Structures for Inductive Link Prediction. *Proceedings of the 32nd ACM International Conference on Multimedia*, 1274–1282.

[18] Xiong, W., Hoang, T., & Wang, W. Y. (2017). DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 564–573. [CrossRef]

[19] Sun, Z., Deng, Z. H., Nie, J. Y., & Tang, J. (2019). RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *International Conference on Learning Representations*.

[20] Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., & Bouchard, G. (2016). Complex embeddings for simple link prediction. *International Conference on Machine Learning*, 2071–2080. [CrossRef]

[21] Yang, F., Yang, Z., & Cohen, W. W. (2017). Differentiable learning of logical rules for knowledge base reasoning. *Advances in Neural Information Processing Systems*, *30*. [CrossRef]

[22] Sadeghian, A., Armandpour, M., Ding, P., & Wang, D. Z. (2019). Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, *32*. [CrossRef]

[23] Tang, X., Zhu, S. C., Liang, Y., & Zhang, M. (2024). RulE: Knowledge Graph Reasoning with Rule Embedding. *Findings of the Association for Computational Linguistics: ACL 2024*, 4316–4335. [CrossRef]

[24] Cheng, K., Ahmed, N., & Sun, Y. (2023). Neural Compositional Rule Learning for Knowledge Graph Reasoning. *The Eleventh International Conference on Learning Representations*.

[25] Lao, N., & Cohen, W. W. (2010). Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, *81*, 53–67. [CrossRef]

[26] Buluta, I., & Nori, F. (2009). Quantum Simulators. *Science*, *326*(5949), 108–111. [CrossRef]

[27] Zhou, Y., Stoudenmire, E. M., & Waintal, X. (2020). What Limits the Simulation of Quantum Computers? *Phys. Rev. X*, *10*(4), 041038. [CrossRef]

Junliang Chen is currently a master student at the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. Her current research interests include knowledge reasoning and quantum reinforcement learning. (Email: 615830571@bupt.edu.cn)

Xianchao Zhang received the Ph.D. degree in Systems Engineering from Beihang University, Beijing, China, in 2013. From 2013 to 2015, he was a Postdoctoral Fellow with Peking University, China. From 2018 to 2022, he was a Postdoctoral Fellow with Southeast University, China. From 2015 to 2021, he was a senior engineer with the China Academy of Electronic and Information Technology. He is currently a Professor with the Provincial Key Laboratory of Multimodal Perceiving and Intelligent Systems and the Engineering Research Center of Intelligent Human Health Situation Awareness of Zhejiang Province, Jiaxing University, China. His research interests include quantum artificial intelligence, intelligent information network, millimeter-wave radar. (Email: zhangxianchao@zjxu.edu.cn)

Fengsong Sun received his bachelor's and master's degrees in Xidian University. He is currently a Ph.D. student in Beijing University of Posts and Telecommunications. His research interests include knowledge graph reasoning and machine learning.(Email: sfs_2021_9@bupt.edu.cn)

Jun Lu received the bachelor's degree from Southeast University in 1985 and the master's degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, in 1988. His research interests include theoretical research and system development in the field of integrated electronic/quantum information systems. He is an Academician of the Chinese Academy of Engineering.(Email: lujun_cetc@163.com)