

RESEARCH ARTICLE

TP-RotateE: A knowledge graph representation learning method combining path information and rules to capture complex relational patterns

Xinliang Liu^{1,2}, Yanyan Shi^{1,2}, Yushi Xu^{1,2}, Yanzhao Ren^{1*}

1 School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing, China, **2** Nation Engineering Research Center for Agri-product Quality Traceability, Beijing Technology and Business University, Beijing, China

* renyanzhao@btbu.edu.cn



OPEN ACCESS

Citation: Liu X, Shi Y, Xu Y, Ren Y (2025) TP-RotateE: A knowledge graph representation learning method combining path information and rules to capture complex relational patterns. PLoS One 20(5): e0324059. <https://doi.org/10.1371/journal.pone.0324059>

Editor: Jiwei Tian, Air Force Engineering University, CHINA

Received: January 6, 2025

Accepted: April 18, 2025

Published: May 27, 2025

Copyright: © 2025 Liu et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data availability statement: The data underlying the results presented in the study are available from <https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding/tree/master/data>.

Funding: This project has received funding support from the National Science and

Abstract

Representation learning on a knowledge graph (KG) aims to map entities and relationships into a low-dimensional vector space. Traditional methods for representation learning have predominantly focused on the structural aspects of triples within the KG. While existing approaches have endeavored to integrate path information and rules to enhance the structural richness of KGs, these efforts have been constrained by the lack of consideration for complex relational representations and contextual information. In this study, we introduce TP-RotateE, an innovative method that leverages the semantic context of triples to effectively capture more intricate relational patterns. Specifically, our model harnesses contextual information surrounding the head entity and distills relevant rules. These rules are then integrated with path information to offer a more holistic perspective on the relationships embedded within complex vector spaces. Furthermore, the synergy between rules and paths empowers the knowledge-embedded model to handle the intricacies of complex relationships. Experimental results on a benchmark dataset confirm that TP-RotateE surpasses current baseline methods in KG inference tasks, achieving state-of-the-art performance.

1. Introduction

A knowledge graph (KG) is a system that gathers and amalgamates information into an ontology and employs a reasoner to deduce new insights [1]. It consolidates various extensive knowledge bases, such as Freebase [2] and AI-KG [3]. Typically, a KG is composed of numerous triples (h, r, t) that encode factual data, where h , r , and t denote the head entity, relation, and tail entity, respectively. Given its capacity to efficiently deliver structured information, KGs are extensively utilized in a range of downstream applications, including intelligent question answering [4], personalized recommendations [5], and information retrieval [6]. However, while large-scale KGs may encompass billions of triples, the knowledge they contain represents only

Technology Major Project (2022ZD0119502). This work was supported jointly by the Beijing Science and Technology Planning Project (Z221100007122003) and the National Key Technology R&D Program of China (No.2021YFD2100605). We gratefully acknowledge the joint research funding from the Beijing Science and Technology Planning Project (Z221100007122003) Project , which enabled us to acquire the advanced computing equipment and relevant data essential for this study. The additional support from the BTBU Digital Business Platform Project by BMEC Project covered the personnel costs for this research.

Competing interests: The authors have declared that no competing interests exist.

a small fraction of the total real-world information and may also harbor inaccuracies and inconsistencies. For instance, 71% of individuals in Freebase lack documented birthplaces, and 75% have no assigned nationality [7]. Recent research has also explored extensively the adversarial robustness of knowledge representations. Tian [8] highlights the potential vulnerabilities in structured knowledge representations that leverage adversarial machine learning techniques to generate covert false data attacks. These studies further illustrate the challenges and progress made in KGR. Such gaps in information can adversely impact downstream tasks. To counter these limitations, the field of knowledge graph reasoning (KGR) has emerged. KGR endeavors to employ reasoning over KGs to surmount existing knowledge gaps and deduce missing elements within the KG.

Recent approaches to knowledge graph completion predominantly rely on embedding techniques, which map entities and relationships into low-dimensional vector spaces [9]. Notable examples in this domain include TransE [10] and TransH [11]. However, single-triple inference methods often struggle with sparse data, while multi-step paths can provide more comprehensive relational information. For instance, research by Seo et al. [12] and Lin et al. [13] seeks to develop more nuanced knowledge representations by integrating intermediate entities and relationship representations along relational paths. Since path representations are intrinsically connected to the relationships, and considering the variety of relationship types in knowledge bases—such as symmetric relationships (e.g., alliances), antisymmetric relationships (e.g., parent-child), inverse relationships (e.g., employer-employee), and compositional relationships (e.g., the head of a company's R&D department)—current methods for embedding relationships find it challenging to fully encapsulate these intricacies. Furthermore, the dependence on numerical computations for path representations also makes them prone to inaccuracies.

To bolster inference precision and enhance interpretability, the incorporation of logical rules is recognized as a potent strategy for enriching sparse knowledge graphs (KGs). Niu et al. [14] and Niu et al. [15] have demonstrated the successful integration of Horn clauses into triples and path embeddings, thereby amplifying the efficacy of representation learning. However, the efficacy of automated rule extraction in sparse KGs is often limited, especially for targeted queries, as the challenge of pinpointing contextually relevant rules remains a formidable obstacle.

In this research, we introduce TP-RotateE, a groundbreaking combinatorial representation learning model that integrates head entity subgraph rules and path information. This model innovatively merges the RotateE embedding approach with path representations, addressing the previous limitations in handling different relationship types and thereby enhancing the precision of relationship embeddings. By harnessing the advanced information interaction capabilities of the Transformer architecture, the model consolidates contextual information around head entities to formulate rules. Once these rules are established in specific formats, they guide the integration of relationships within paths. The TP-RotateE model was subjected to rigorous evaluation across three benchmark datasets. The experimental outcomes not only highlight its exceptional performance in KG completion tasks but also significantly outperform

existing baselines, substantiating the efficacy of integrating rules and paths in KG embeddings. The pivotal contributions of this study are as follows:

- (1) The incorporation of subgraph rule information centered on head entities, which facilitates the aggregation of subgraph information and the exploitation of contextual cues to bolster the reasoning process and improves the model's ability to capture relational structures.
- (2) The synergy between RotatE embedded models and paths achieves a more comprehensive representation of relationship types, which enables a more expressive representation of different types of relationships and allows for the merging of more complex relationships. This integration not only captures the various relationship patterns more effectively, but also facilitates their combination to model more complex relationship structures.
- (3) The proposal of a knowledge representation learning model that seamlessly integrates relationship path information, logical rule information, and entity and relation information from the RotatE embedding model. Our experiments confirm that this model outperforms all baseline methods, setting a new standard in the field.

2. Related work

Embedding model based on translation mechanism: In recent years, significant advancements have been made in distributed representations of entities and relations in KG learning, which can be categorized into four primary groups: 1) **Traditional translation embedding models.** Inspired by the word word vector embedding model Word2Vec, TransE interprets relations as translation operations in a low-dimensional embedding space. Specifically, for a valid triple (h, r, t) , the vector of the head entity combined with the relation vector should approximate the vector of the tail entity, satisfying $h + r \approx t$. However, TransE performs poorly in handling complex relations such as one-to-many or many-to-many. The TransH model addresses this limitation by introducing projection, allowing entities to be represented differently under specific relation types. 2) **Translation embedding models based on neural network.** DistMult [16] simplifies matrix decomposition by restricting bilinear matrices to diagonal matrices. By introducing tensor decomposition in ComplEx [17] for knowledge completion, ComplEx effectively handles numerous binary relations, overcoming the limitation of DistMult in representing symmetric relations. Additionally, Wang [18] utilizes Capsule Networks to model knowledge graphs, capturing hierarchical entity dependencies and improving robustness to noisy data. By leveraging dynamic routing mechanisms, Caps-OWKG effectively represents relational structures and enhances knowledge graph completion performance. 3) **Translation embedding models based on relational paths.** In KGs, entities are connected through indirect relations, which are addressed by multi-step path inference models. While TransE only considers direct relationships, it fails to capture relational paths. PTransE [19] model treats relational paths as transformations between entities, employing the path constraint resource allocation algorithm to evaluate path reliability and representing paths through the recursive combination of relational embeddings. Additionally, Jin [20] explores how rule-based relational patterns can enhance knowledge graph embeddings. By leveraging learned logical rules, this approach captures multi-hop relational dependencies, improving the expressiveness of translation-based models in complex reasoning tasks. Recently, Dong [21] introduces a memory mechanism to dynamically update path representations over time, enhancing temporal knowledge graph reasoning. This approach improves the model's ability to capture time-dependent multi-hop relations, making it more effective for dynamic knowledge graphs. 4) **Translation embedding models based on geometric relations.** TransF [22] models the relationship between the head and tail entities as a flexible vector translation, avoiding strict transformations based on vector addition. For triples (h, r, t) , it relaxes the requirement of $h + r \approx t$, instead enforcing $h + r \approx at$, where a reflects flexibility. This approach maintains consistent embedding directions without requiring vector magnitudes, improving model performance in reflexive, one-to-many, and many-to-one relations.

Despite these significant strides, these methods face limitations due to their exclusive reliance on triplet information. This dependency can result in limited accuracy and diminished interpretability.

Rule extraction and models enhanced by rules: The interpretability and semantic richness of logical rules render them indispensable in knowledge reasoning. They empower systems to engage in more profound reasoning and comprehension, thereby enhancing the quality and utility of knowledge graphs (KGs). In the realm of rule learning on KGs, the rule mining method employed by AMIE [23] aims to generate edge relationship rules. It initiates by pre-generating all potential rules predicated on edge types. Subsequently, it discerns instances within the graph that conform to these rules and computes their confidence levels. A rule is deemed valid if its confidence surpasses a predefined threshold. RuleN [24], on the other hand, proposes an efficient framework for rule mining. It systematically identifies paths of length n between the head entity a and the tail entity b using depth-first search. These paths serve as the rule bodies, which are then utilized to construct the rules.

The essence of traditional symbol-based rule learning methods lies in approximating specific paths derived from graph traversal as rules. This is achieved by searching and exploring paths across the entire graph or a sampled subset. The rule generation process in these algorithms is significantly influenced by both the search algorithms employed and the pruning techniques applied. The choice and optimization of these methods are pivotal for ensuring the quality and efficiency of the rule mining results.

Recently, differentiable rule-learning methods have gained popularity for their ability to simultaneously learn the confidence and structure of rules. Neural LP [25], based on Tensorlog [26], integrates first-order parameter learning with the structural learning of logic rules in an end-to-end microscopic model. Neural-num-LP [27] extends Neural LP by incorporating numerical attributes into rule bodies, enabling operations such as comparisons, aggregations, and negations between entities. Meanwhile, in the field of deep learning, Saeedan et al. [28] proposed Detail-Preserving Pooling (DPP), which enhances traditional pooling methods by preserving critical details while reducing feature dimensions.

In KGs with sparse data, recent studies have explored integrating rules with KG embeddings to enhance data augmentation. Cohen et al. [29] utilized relational embeddings to derive rules and generate new factual triples associated with sparse entities based on these inferred rules. However, this method generates only a limited set of rules due to the restricted relationships in the KG and fails to exploit its inherent semantic information. Li et al. [30] proposed a logically guided semantic representation learning model for zero-sample relation classification. The method establishes implicit and explicit semantic connections between seen and unseen relations through knowledge graph embeddings and logic rules, effectively bridges the gap between seen and unseen relations, and shows significant improvement potential in zero-sample relation classification tasks. Wang et al. [31] enriched the model by incorporating ontological information of entities, enhancing the ability to distinguish between multiple entity classes and improving both accuracy and interpretability. Previous rule mining approaches, which relied on automated rule mining, were inadequate for addressing context-aware information in KGs, an aspect crucial for handling sparse KGs.

3. Methodology

This study seamlessly integrates subgraph-based rule mining with relational paths to distill more profound semantic insights from the model. The comprehensive framework of the model is depicted in Fig 1. The process first mines rules of varying lengths based on the contextual information surrounding the head entity. R_1 and R_2 represent rules of length 1 and 2, respectively. After that, combined with the paths mined from the knowledge graph, rule R_2 is used to synthesize the paths while rule R_1 establishes semantic links between specific relational formulas. Concurrently, entities and relations are projected into a vector space through vector initialization, facilitating the training of embeddings for the KG. The learning process is designed to refine the representation of triples, paths, and correlation pairs, thereby enhancing their semantic cohesion.

3.1. Rule information extraction

The model is composed of two core components: 1) An encoder block that assimilates subgraph information related to the head entity and facilitates the interaction of contextual information. This component is crucial for capturing the rich

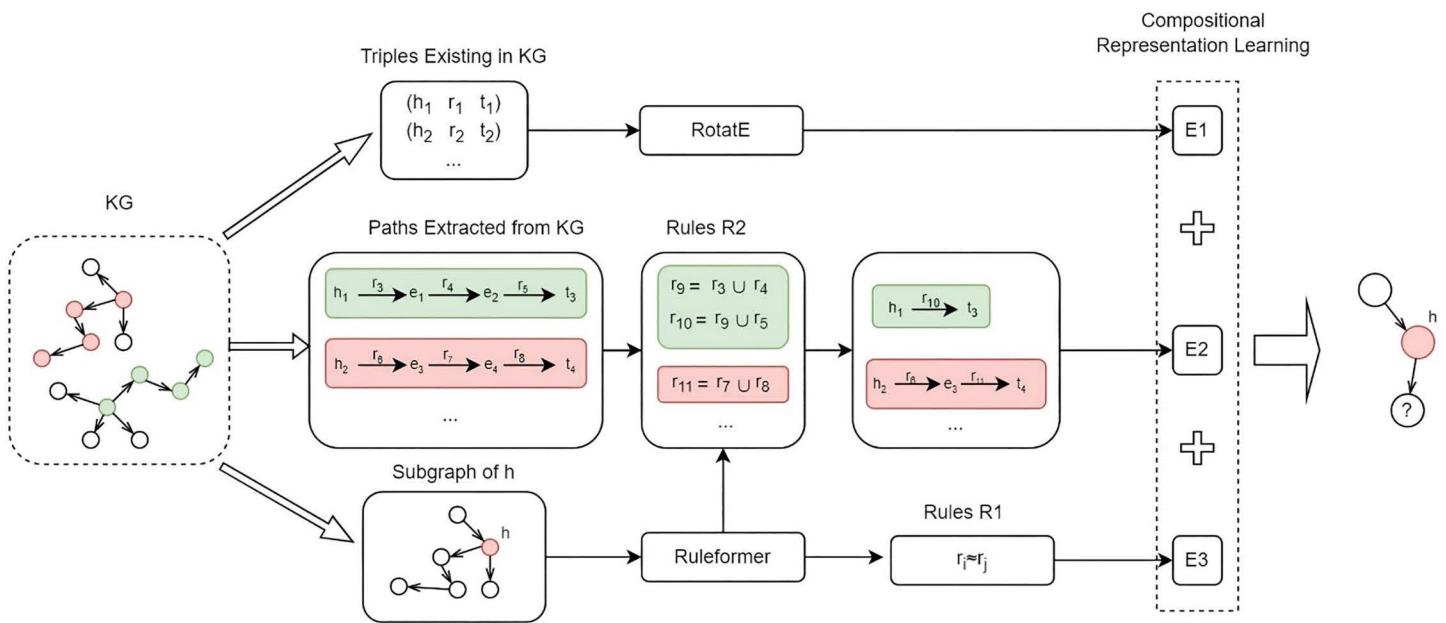


Fig 1. Overall architecture of the model.

<https://doi.org/10.1371/journal.pone.0324059.g001>

contextual nuances that surround the head entity within the knowledge graph; 2) A decoder block that leverages the aggregated entity embeddings to calculate relational probabilities at each stage of rule micromining, as illustrated in Fig 2. This component plays a pivotal role in translating the aggregated information into actionable insights, particularly in the context of rule-based reasoning within the knowledge graph.

For triples (h, r, t) , the correlation subgraph of the head entity h is first extracted, as it potentially contains contextual information about the head entity h . $S_k(h)$ is defined as the subgraph consisting of the k -hop neighbors of h along with the set of edges connecting these entities. Since $S_k(h)$ is a graph structure and the Transformer model operates as a seq2seq framework, it is necessary to transform the subgraph into a sequence, denoted as $S_{node} = [e_1, e_2, \dots, e_{num}, \dots, blank]$, where num represents the total number of entities surrounding the head entity, and a token $blank$ is needed for padding. The positional embedding is then derived by appending the shortest path distance to the central entity h .

To enhance representation learning, the type of relationship associated with the entity is incorporated. The representation x_e of an entity e is computed by summing the type embedding of the associated relationship with the randomly initialized embedding y_e . This process is expressed as follows:

$$x_e = \sum_{i=1}^{|R|} b_i^{dom} r_i^{dom} + \sum_{i=1}^{|R|} b_i^{ran} r_i^{ran} + y_e \quad (1)$$

Here, $rdom$ and $rran$ denote the domain and range embeddings of the relation r_i , respectively. The parameters $bdom$ and $bran$ are normalized to account for the diversity of distinct relation types. This normalization is expressed as follows:

$$b_i^{dom} = \frac{n_i^{dom}}{\sum_{j=1}^{|R|} n_j^{dom}}, b_i^{ran} = \frac{n_i^{ran}}{\sum_{j=1}^{|R|} n_j^{ran}} \quad (2)$$

where $ndom$ and $nran$ represent the number of relationships of each type connected to or associated with e . Accordingly, the sequence of nodes in the subgraph can be expressed as $S_e = [x_1, x_2, \dots, x_{num}, \dots, blank]$. Building on this, attention

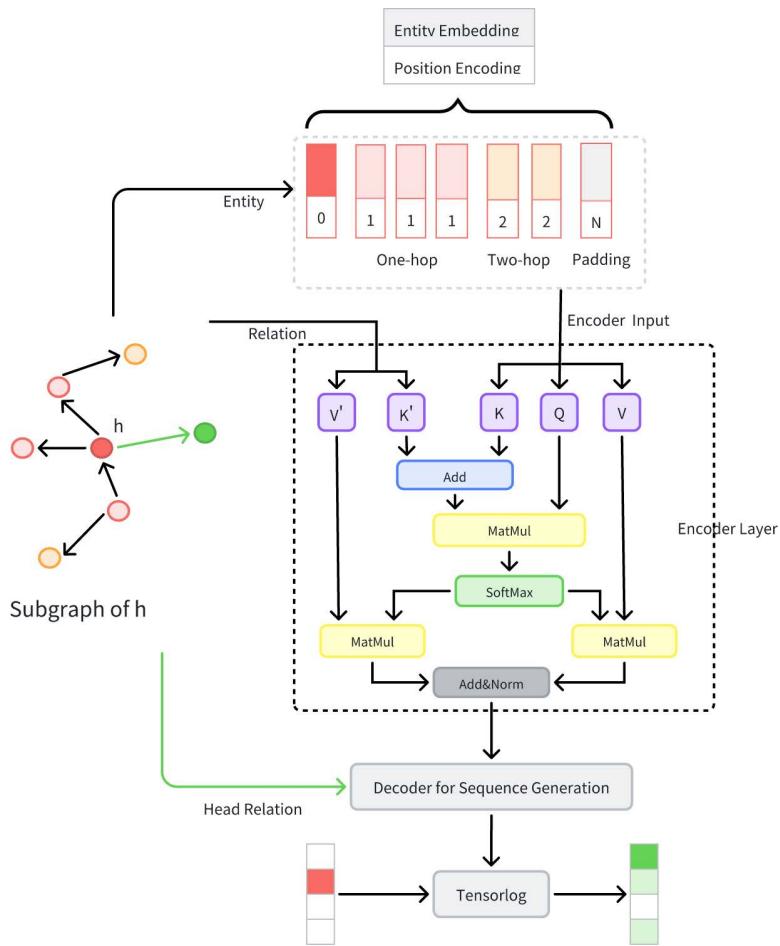


Fig 2. Model framework for rule extraction.

<https://doi.org/10.1371/journal.pone.0324059.g002>

mechanisms are introduced to incorporate edge information. Similar to embedding relational information into entity representations, relationship data are integrated into the value computation step:

$$z_i = \sum_{j=1}^n a_{ij} \left(x_j W^V + \sum_{r=1}^{|R|} k_r x_r W^{V'} \right) \quad (3)$$

where W^V represents the entity value matrix, and $W^{V'}$ denotes the relational value matrix.

After the encoder outputs the entity sequence S'_e , the decoder combines the encoded information with the head entity and iteratively generates the sequence until the rule sequence S_r with a rule length of T is produced. This process constitutes the rule mining performed by the model.

Specifically, the head relation x_r is embedded as the starting element of the rule sequence S_r in the decoder input, i.e., $S_0 = x_r$. Through cross-computation of S_r and S'_e , the decoder obtains the vector for the subsequent relation. After MLP deployment, the probability ω_{rt} corresponding to relation r_t at step t is derived. During inference, the relation with the highest probability is selected to complete the remaining sequence.

$$w_{t+1} = \text{MLP}(\text{CrossAttention}(S_r^t, S'_e)) \quad (4)$$

In the inference process, $\omega_{t+1} \in R^{|R| \times 1}$ denotes the probability distribution over all possible relations at step t . Let r_{t+1} represent the relation with the highest probability among them. Based on this, $S_r^{t+1} = [S_r^t, x_{r_{t+1}}]$ can be determined. This procedure is iterated T times to construct a complete rule set of length T .

Using this method, rules of lengths 1 and 2 are derived and encapsulated into the respective rule sets, denoted as R_1 and R_2 .

For the rule set of length 1, it is assumed that when the rule holds true, the semantic value of the relation r_i in the rule set is greater than that of the direct relation r_2 between entities. Furthermore, the embedding representation of a pair of relations in a rule set exhibits higher semantic similarity than that of two non-contiguous relations. It should be noted that in representation learning, the rule $(a, r_2, b) = (b, r_1, a)$ should be represented as $(a, r_2, b) = (a, r_1^{-1}, b)$.

For the rule set of length 2, the previous rule generation model is applied to derive rules for the following pattern:

$$r(A, B) \leftarrow r_1(A, C_1) \wedge \dots \wedge r_T(C_{T-1}, B)$$

Let T denote the length of the rule. Here, r and r_i represent relations within the set R . The variables A , B , and C , act as placeholders that can be substituted with specific entities. The expression $r(A, B)$ denotes a rule in the form of a triple. The segment of the triple to the left of the arrow is referred to as the rule head, while the segment to the right is identified as the rule body. For the rule set R_2 , the rules are encoded to represent a directed path corresponding to the rule body. This path is sequentially constructed from the atoms of each rule body. By encoding these eight rules, a valid path set $P(h, t)$ is established for the entity pair (h, t) . The encoding rules are shown in [Table 1](#).

To harness the encoded rules effectively, paths must be traversed semantically, iteratively combining operations until no further relationships can be merged. This process encompasses two primary scenarios: (1) In the ideal case, all relationships along the path can be sequentially combined using rule R_2 , culminating in a single relationship that connects the entity pairs. (2) More commonly, some relationships cannot be directly formed based on rule R_2 , necessitating numerical operations, such as addition, to embed these relationships into the model. Furthermore, when multiple rules can be matched along a path simultaneously—for instance, when both $R(a, c) \wedge T(c, b) \Rightarrow U(a, b)$ and $R(a, c) \wedge T(c, b) \Rightarrow V(a, b)$ are activated—the rule with the highest confidence is selected to guide the combination of relationships. This approach ensures that the most reliable rule influences the embedding process, thereby enhancing the accuracy of the knowledge graph.

3.2. Relational path modeling

In this section, the model component utilizing paths is introduced, focusing on the representations of entities and relationships encompassing three types of relations. For a KG comprising a set of triples $S = \{(h, r, t)\}$, the composite embedding capabilities

Table 1. R_2 lists the transformation modes.

Original Rules ^a	Encoder Rules ^b
$r_3(a, b) = r_1(a, e) \cup r_2(e, b)$	$r_3 = r_1 \cup r_2$
$r_3(a, b) = r_1(e, b) \cup r_2(a, e)$	$r_3 = r_2 \cup r_1$
$r_3(a, b) = r_1(e, b) \cup r_2(e, a)$	$r_3 = r_{-12} \cup r_1$
$r_3(a, b) = r_1(e, a) \cup r_2(e, b)$	$r_3 = r_{-11} \cup r_2$
$r_3(a, b) = r_1(a, e) \cup r_2(b, e)$	$r_3 = r_2 \cup r_{-12}$
$r_3(a, b) = r_1(b, e) \cup r_2(a, e)$	$r_3 = r_2 \cup r_{-11}$
$r_3(a, b) = r_1(e, a) \cup r_2(b, e)$	$r_3 = r_{-11} \cup r_{-12}$
$r_3(a, b) = r_1(b, e) \cup r_2(e, a)$	$r_3 = r_{-12} \cup r_{-11}$

^aThe left side shows the original rule extracted from the KG.

^bThe right side shows the encoded rule.

<https://doi.org/10.1371/journal.pone.0324059.t001>

of the head entity h and the tail entity t are mapped, where $h, t \in C_k$, r denotes the relationship linking the two entities. When the relationship is valid, the model is designed to yield a low energy score; otherwise, a high energy score is expected.

For every triplet (h, r, t) , Rotate represents the element-wise rotation induced by the relation r , mapping the head entity h to the tail entity t . The scoring function is defined as:

$$E(h, r, t) = \| h \circ r - t \| \quad (5)$$

Here, \circ represents Hadamard product. For sparse data, the training samples in the KG are often limited, making it difficult for models to accurately identify precise connections between them. Incorporating path information enables the linkage of long-tail entities and relationships with others, thereby facilitating the extraction of richer and more accurate inferences from the data.

The path between entities h and t is represented as $p(r_1, r_2, \dots, r_n)$, and there may be multiple such paths. The triplet score $E(h, P, t)$, which accounts for multi-step relationships, is defined as:

$$E(h, P, t) = \frac{1}{Z} \sum_{p \in P(h, t)} R(p|h, t) E(h, p, t), \quad (6)$$

$R(p|h, t)$ represents the reliability of the connection pathway between entities h and t . $Z = \sum_{p \in P(p|h, t)} R(p|h, t)$, acting as a normalization coefficient, is combined with $E(h, p, t)$, which denotes the energy function of the triplet (h, p, t) .

The Path-Constrained Resource Allocation (PCRA) algorithm is employed to evaluate the reliability of relational pathways. Specifically, for the path triplet (h, p, t) , the pathway $p = (r_1, r_2, \dots, r_i)$ from the starting entity h to the terminal entity t is determined, and the pathway can be expressed as $E_0 \xrightarrow{r_1} E_1 \xrightarrow{r_2} \dots \xrightarrow{r_i} E_t$, where $h \in E_0$, and $t \in E_t$. For any entity $m \in E_i$, its preceding edge along relation r_i is denoted as $B_i(m)$, and its succeeding edge along relation r_i is represented as $C_i(m)$. The resources directed to m are defined as follows:

$$R(m) = \sum_{n \in B_i(m)} \frac{1}{|C_i(n)|} R(n) \quad (7)$$

where $R(n)$ represents the resources derived from entity n . For each relational path, the initial resource is set as $R(h) = 1$, and resources are iteratively allocated along the path to determine the resource allocation $R(t)$ of the tail entity. The reliability of the relational path is represented by the resource obtained by the tail entity, $R(t) = R(p|h, t)$.

To compute the energy function of the path triplet (h, p, t) , a method analogous to the scoring function in the RotatE triplet is employed. For the path $p = (r_1, \dots, r_i)$, the composite operation is defined as follows to obtain the path embedding.

$$p = r_1 + \dots + r_i \quad (8)$$

3.3. Compositional representation modeling

For every triplet (h, r, t) , this study defines the following energy functions, which establish modular dependencies for triplets of path pairs based on rules R_2 and R_1 .

$$E_1(h, r, t) = \| hr - t \|, \quad (9)$$

$$E_2(p, r) = R(p|h, t) \left(\prod_{\mu_i \in U(p)} \mu_i \right) \| H(p) - r \|, \quad (10)$$

$$E_3(r, r_e) = \| r - r_e \|, \quad (11)$$

The energy function E_1 represents the triplet score. E_2 denotes the energy function used to evaluate the similarity between path p and relation r . The confidence level set $U(p) = \mu_1, \dots, \mu_n$ corresponds to all rules in the rule set R_2 applied during the formation of path p . $E_3(r, r_e)$ characterizes the relationship between relation r and r_e . If r_e can be associated with relation r in the triplet through rule R_1 , E_3 is expected to assign it a lower value.

3.4. Loss function

The loss function is defined as a fractional function based on margins, specifically designed for the purpose of negative sample sampling.

$$L = \sum_{(h,r,t) \in T} \sum_{(h',r',t') \in T'} \left(L_1(h, r, t) + \alpha_1 \sum_{p \in P(h,t)} L_2(p, r) + \alpha_2 \sum_{r_e \in R(r)} L_3(r, r_e) \right), \quad (12)$$

T represents the positive triples, while T' denotes the corresponding negative samples. $R(r)$ includes relations derived from r through rule R_1 . $P(h, t)$ represents the set of paths between (h, t) , with P being an individual path. L_1 , L_2 , and L_3 correspond to the marginal loss functions for entity triples (h, r, t) , path-relation pairs (p, r) , and relation pairs (r, r_e) , respectively. Specifically, the marginal loss function for entity triples is given as follows:

$$L_1(h, r, t) = \max \left(0, -\log \sigma(\gamma_1 - E(h, r, t)) - \sum_{i=1}^n \frac{1}{k} \log \sigma(E(h', r, t') - \gamma_1) \right) \quad (13)$$

where γ_1 represents a fixed boundary value, and σ denotes the sigmoid function. During the triplet negative sampling process, a self-adversarial negative sampling approach is employed to address the inefficiency associated with standard negative sampling. Specifically, negative triples are sampled as follows:

$$p(h'_j, r, t_j | \{(h_i, r_i, t_i)\}) = \frac{\exp f_r(h'_j, t_j)}{\sum_i \exp f_r(h'_i, t_i)}, \quad (14)$$

α represents the sampling temperature. To reduce costs, probabilities are employed as weights for negative samples. The final loss function is defined as follows:

$$L_1(h, r, t) = \max \left(0, -\log \sigma(1 - E_1(h, r, t)) - \sum_{i=1}^n p(h', r, t') \log \sigma(E_1(h', r, t') - 1) \right), \quad (15)$$

$$L_2(p, r) = \max(0, \gamma_2 + E_2(p, r) - E_2(p', r')), \quad (16)$$

$$L_3(r, r_e) = \max(0, \gamma_3 + \beta E_3(r, r_e) - E(r, r')). \quad (17)$$

where γ_2 and $\gamma_3 > 0$ are hyperparameters, and β represents the confidence level associated with r and r_e . The Adam optimizer [26] was employed, and hyperparameters were fine-tuned on the validation set. To maintain training efficiency, the path length was restricted to a maximum of 3 steps.

4. Experiments

4.1. Experimental setting

Datasets and rules. The proposed model was evaluated on three benchmark datasets: FB15K and FB15K-237 from Freebase, and WN18 from WordNet. FB15K-237, a subset of FB15K, excludes inverse relations and emphasizes

modeling symmetric/antisymmetric properties and compositional patterns for link prediction. [Table 2](#) provides detailed dataset statistics, including the number of entities, relationships, and triples in the training, validation, and testing sets. The performance of the proposed method was compared with baseline approaches on entity prediction tasks for KG completion.

Evaluation protocols. Test triples, along with candidate triples not included in the training, validation, or test sets, were organized for evaluation. The candidate triples were generated by modifying the subject or object, yielding (h', t, r) or (h, t', r) . The primary metrics used for evaluation are Mean Rank (MR), Mean Reciprocal Rank (MRR), and Hits@ n , which represents the proportion of correct entities ranked in the top n predictions. The goal is to achieve a lower MR, higher MRR, and higher Hits@10. To accomplish this, reconstructed triples are scored using the following function:

$$Q(h, r, t) = \|hr - t\| + \alpha_1 \sum_{p \in P(h, t)} R(p | h, t) \left(\prod_{\mu_1 \in B(p)} \mu_1 \right) \|C(p) - r\| \quad (18)$$

As shown in [Eq. 18](#), rule R_2 is utilized to amalgamate paths during the testing phase.

Baselines. For the purpose of evaluating the efficacy of our approach, a selection of models for knowledge graph completion serves as benchmarks. These benchmarks are divided into three distinct categories: (1) embedding models that concentrate exclusively on triples, such as TransE, DistMult, ComplEx, and RotatE; (2) a path-based model, PTransE; and (3) a rule-enhanced model, RPJE. We have adopted the optimal results reported in their respective original studies and implemented RotatE and RPJE using their authentic source codes.

Experimental settings. To ensure a fair and equitable comparison, the following configurations were employed: (1) The real and imaginary parts of the entity embeddings were initialized randomly, whereas relation embeddings were initialized uniformly across the interval from 0 to 2π . Regularization was not included, as a fixed margin γ_1 effectively mitigated overfitting. In line with the standard baseline setups, the learning rate was set to 0.0001, and γ_2 was fixed at 1.

A furthermore, a grid search was performed to determine the most effective hyperparameters. These included embedding dimensions d spanning from 125 to 1000. Furthermore, a grid search was performed to determine the most effective hyperparameters. These included embedding dimensions d spanning from 125 to 1000, self-adversarial sampling temperatures α of 0.5 and 1.0, a fixed margin γ_1 set at 3, and γ_3 values ranging from 0.5 to 3.5. We utilized the best results reported in the original studies and implemented RotatE and RPJE using their respective source codes.

4.2. Quality assessment of rules

In the realm of rule mining, Standard Confidence (SC) is a widely adopted metric for assessing the quality of extracted rules., measuring how often the rule head occurs if the rule body holds. In order to fairly compare the rule quality of different methods, we compute the average SC scores of each method on the top K ($K=50, 100, 200, 500$) rules and conduct experiments on the FB15K-237 dataset (as shown in [Table 3](#)). The rules for NeuralLP [32] and DRUM [33] were derived from their original implementations. Taking the FB15K-237 dataset as an example, the table presents the results of various rule extraction models. The findings reveal that Ruleformer achieves a significantly higher standard confidence for subgraph-based parsing rules compared to other methods when dealing with triplets that share the same relationship.

Table 2. Data statistics of the three datasets used in the experiment.

Dataset	Relationships	Entities	Training	Validation	Testing
FB15K	1,345	14,951	483,142	50,000	59,071
FB15K-237	237	14,541	272,115	17,535	20,466
WN18	18	40,943	141,442	5,000	5,000

<https://doi.org/10.1371/journal.pone.0324059.t002>

Higher confidence levels indicate that more reliable and valid rules are being utilized in the process. Additionally, it has been observed that under the same model, rules with a length of 2 are more effective, likely because longer paths often lead to less accurate compositions. As a result, a path step size of 2 was chosen as the optimal configuration for the subsequent results.

4.3. Link prediction

Link prediction is a cornerstone task in knowledge graph (KG) embedding, focused on forecasting missing or potential connections between entities within a KG. It is pivotal for KG completion, knowledge base reasoning, and a variety of applications that necessitate inference or the uncovering of missing facts. A comparative analysis was performed between RotatE and several leading models, with RotatE and RPJE being chosen as benchmarks. Their published results were directly utilized due to the common evaluation dataset employed. [Table 4](#) displays our findings on the FB15k, WN18, and FB15K-237 datasets, clearly showing that TP-RotatE surpasses all other cutting-edge models in performance.

The proposed TP-RotatE method was initially subjected to a rigorous evaluation against various baselines for the task of link prediction on the FB15K dataset. The following insights can be gleaned from the results presented in [Table 4](#): (1) TP-RotatE outperforms the baseline models, with the majority of these improvements being statistically significant, demonstrating its robustness and reliability. (2) Notably, TP-RotatE outshines RPJE across all evaluated metrics, signifying its superiority in leveraging head entity rules extracted via a neural network approach. This leads to higher accuracy in path composition and enhanced path embeddings. (3) TP-RotatE shows a 6.3% and 4.4% improvement in Mean Reciprocal Rank (MRR) compared to the rule-based baseline RotatE. This underscores the effectiveness of incorporating rules to preserve more semantic information and to strengthen the integration of paths. Furthermore, experiments conducted on the FB15K-237 dataset reveal that while TP-RotatE performs comparably to RPJE, it exhibits slight advantages on most metrics. This suggests that TP-RotatE is more refined and efficient in integrating paths and rules. On the FB15K-237 dataset, where inverse relationships are not present, efficient rule mining serves as a valuable adjunct to link prediction, bolstering semantic relevance.

4.4. Experimental results by relation category

In knowledge graph (KG) link prediction, the diversity of relationship types encapsulates specific interactions and semantic linkages between entities, which significantly impact the precision of predictions. Therefore, the analysis of link prediction outcomes is segmented according to these varying relationship types. As defined by Bordes et al. [10], relationships within a knowledge base can be categorized based on their mapping characteristics (1-1, 1-N, N-1, N-N). [Table 5](#) details

Table 3. Avg. confidence on FB15K-237, by rule length T.

Methods	FB15K-237		
	TOP		
	50	200	500
Neural-LP($T^*=2$)	.020	.044	.033
Neural-LP($T=3$)	.020	.031	.034
DRUM($T=2$)	.058	.036	.048
DRUM($T=3$)	.020	.039	.027
Ruleformer($T=2$)	.241	.338	.310
Ruleformer($T=3$)	.313	.322	.282

*T represents the depth of the inference step.

<https://doi.org/10.1371/journal.pone.0324059.t003>

the comparative performance of our model and several selected baselines across different types of relationships, with the evaluation conducted using the FB15K dataset.

- 1) The proposed model demonstrates a consistent and significant outperformance across all baseline methods within various relational categories. It achieves an average improvement of 2.1% in head entity prediction compared to RPJE, with particularly remarkable enhancements of 3.6% observed in many-to-one and many-to-many relationships. In the case of tail entity predictions, the model shows an average improvement of 1.8%, with more substantial gains of 2.4% in many-to-one and many-to-many scenarios.
- 2) These results suggest that the model is equally effective on non-injective relationships as it is on one-to-one relationships. This highlights the significance of integrating additional entity rules to retain semantic information, which markedly improves the accuracy of entity prediction.

4.5. Ablation study

To assess the importance of TP-RotateE components, ablation experiments were conducted on FB15K by removing paths and extracted rules. As shown in [Table 6](#), tp-RotateE-p and tp-RotateE-rul represent the TP-RotateE model without paths and without logical rules, respectively. Evidently, the removal of any component results in a decline in model performance.

Table 4. Link prediction results on FB15k, WN18, and FB15K-237.

Models	FB15K			WN18			FB15K-237		
	MR	MRR	H@10	MR	MRR	H@10	MR	MRR	H@10
TransE	—	.463	.749	—	.495	.943	357	.294	.465
DistMult	42	.798	.893	655	.797	.946	254	.241	.419
ComplEx	—	.692	.840	—	.941	.947	339	.247	.501
PTTransE	54	.679	.834	—	.890	.945	302	.363	.526
RPJE	40	<u>.816</u>	<u>.903</u>	—	.946	.951	207	<u>.470</u>	<u>.625</u>
RotatE	40	.797	.884	309	<u>.949</u>	<u>.959</u>	<u>177</u>	.338	.533
TP-RotateE	42	.827	.912	318	.951	.960	169	.484	.642

*Missing values indicate unreported scores.

Bold highlights the best score, underline marks the second best.

<https://doi.org/10.1371/journal.pone.0324059.t004>

Table 5. Results on FB15K.

Models	Head Prediction (Hits@10)				Tail Prediction (Hits@10)			
	1-1	1-N	N-1	N-N	1-1	1-N	N-1	N-N
TransE	.437	.657	.182	.472	.437	.197	.667	.500
TransH	.668	.876	.287	.645	.655	.398	.833	.672
ComplEx	.939	.969	.692	.893	.938	.823	.952	.910
TransG	.930	.960	.625	.868	.928	.681	.945	.888
PTTransE	.910	.928	.609	.838	.912	.740	.889	.864
RPJE	<u>.942</u>	.965	<u>.704</u>	<u>.916</u>	<u>.941</u>	<u>.839</u>	.953	<u>.933</u>
RotatE	.922	<u>.967</u>	.602	.893	.923	.713	<u>.961</u>	.922
TP-RotateE	.949	.971	.743	.931	.950	.852	.970	.961

<https://doi.org/10.1371/journal.pone.0324059.t005>

4.6. Case study

Consider an entity prediction task as shown in Fig 3, given that the head entity is *Eiffel Tower* and the relation is *localiseIn*. Our goal is to predict the tail entity. Our model TP-RotatE yields the result: *France*. Here is a detailed description of the reasoning process.

For simple two-step paths, the embedding model in the model can directly activate '*locatedIn(x, y) = (situatedIn(x, z) ∧ capitalof(z, y))*' by successive rotation actions, which in turn predicts the tail entity *France*.

For the three-step path of the complex relationship, the model first determines the association of *Eiffel Tower* with *FrenchCulture* by the rule '*CulturalSignificance (architecturalStyle, associatedWith)*', and then determines the association of *Eiffel Tower* with *FrenchCulture* by the rule '*locatedIn (FrenchCulture, France)*' determines that *FrenchCulture* is located in *France*. This intermediate combination result *CulturalSignificance* is further used to achieve the predicted result *France* for the tail entity through embedding on the reconstructed path of the inclusion relation *FrenchCulture*. The three-step path effectively simplifies the combination of complex relations through the model's path fusion technique, and rule mining also provides more possibilities for combining paths.

4.7. Model performance analysis on different types of relationships

To conduct an in-depth analysis of how various models handle different relationship types—including symmetric, antisymmetric, and compositional relations—we evaluate their performance using Mean Reciprocal Rank (MRR) and Hits@10. The results, visualized in bar charts (Figs 4 and 5), offer a comprehensive comparison across models. Below, we summarize the key findings: Across all evaluation metrics, TP-RotatE consistently outperforms other models, achieving an MRR of 0.951 and H@10 of 0.96. These results highlight its superior capability in the knowledge graph completion task. As shown in Fig 4, TP-RotatE maintains a leading position in modeling symmetric relations (MRR=0.971), surpassing RotatE (0.968) and RPJE (0.966). This suggests that TP-RotatE effectively captures relational equivalence, making it particularly

Table 6. Ablation studies by removal paths and logical rules.

Models	MR		MRR	Hits@10%	
	Raw	Filtered	Filtered	Raw	Filtered
TP-RotatE-p	186	40	0.811	50.2	88.4
TP-RotatE-rul	217	49	0.714	44.3	81.8
TP-RotatE	183	38	0.824	51.4	90.9

<https://doi.org/10.1371/journal.pone.0324059.t006>

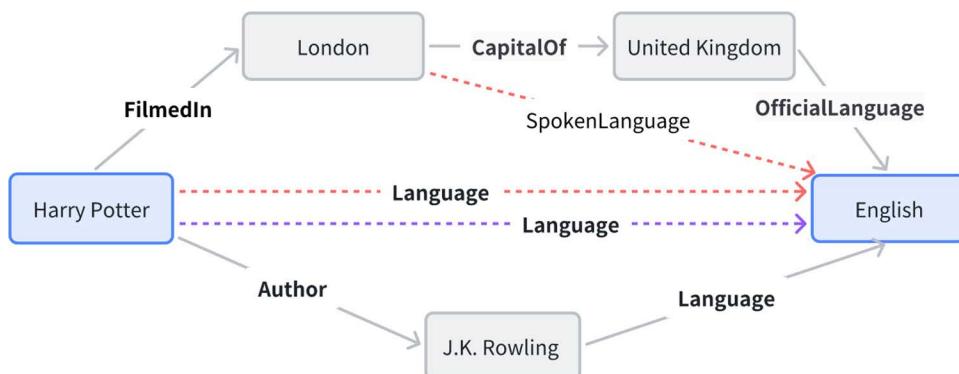


Fig 3. An example of the explainable entity prediction.

<https://doi.org/10.1371/journal.pone.0324059.g003>

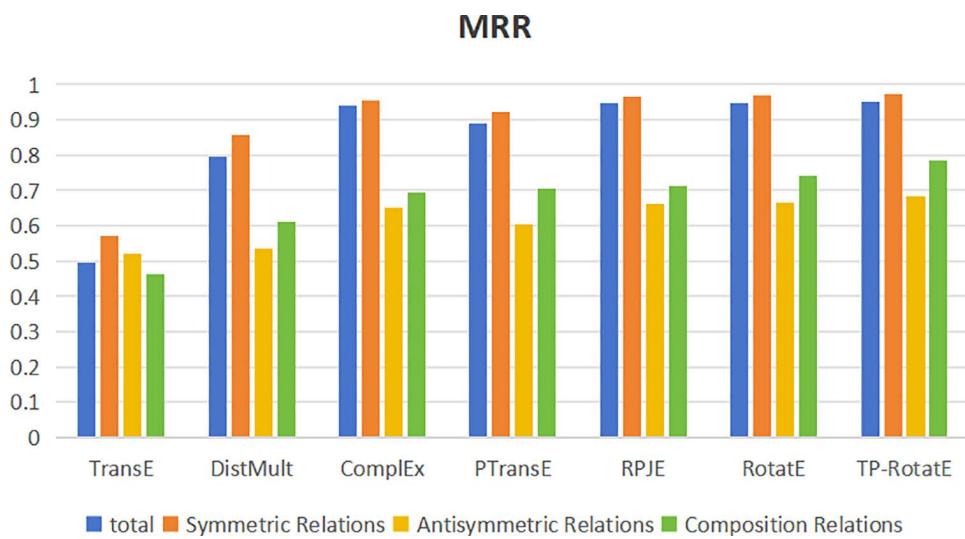


Fig 4. MRR comparison across different relationship types.

<https://doi.org/10.1371/journal.pone.0324059.g004>

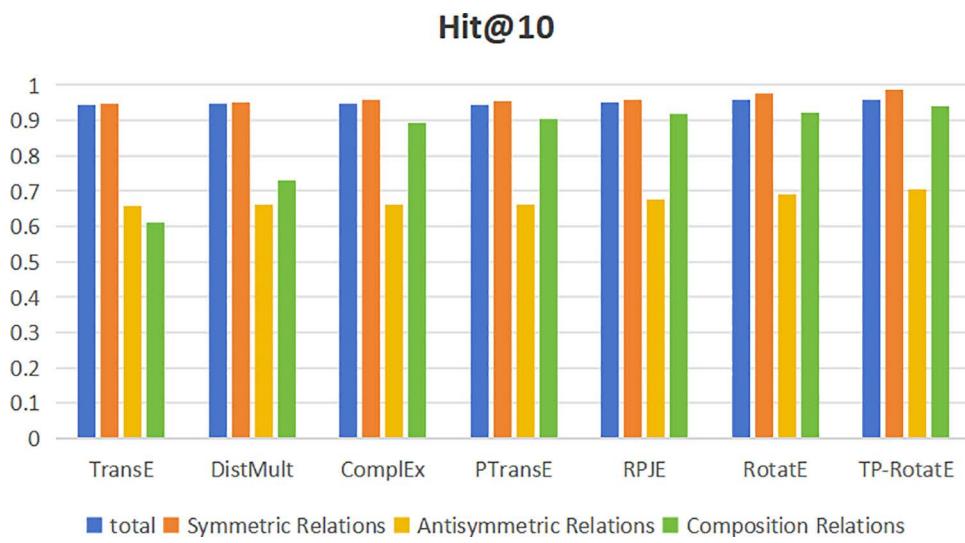


Fig 5. Hit@10 comparison across different relationship types.

<https://doi.org/10.1371/journal.pone.0324059.g005>

well-suited for learning symmetric dependencies. For antisymmetric relations, TP-RotateE demonstrates a clear advantage with an MRR of 0.683, outperforming RotatE (0.664) and ComplEx (0.649). These results indicate that TP-RotateE is more adept at distinguishing directional dependencies, thereby enhancing predictive accuracy when modeling antisymmetric relations. When handling compositional relationships, TP-RotateE once again achieves the highest performance (MRR = 0.786), outperforming RPJE (0.713) and RotatE (0.741). This suggests that TP-RotateE exhibits superior generalization capabilities in learning complex relational structures.

As depicted in Fig 5, the H@10 trends align closely with the MRR results, further reinforcing TP-RotateE's superior adaptability across different relationship types. These findings suggest that TP-RotateE is not only effective in general knowledge graph completion but also excels at modeling diverse relational structures. The empirical analysis

demonstrates that TP-RotatE consistently achieves state-of-the-art performance across symmetric, antisymmetric, and compositional relations. These results underscore its enhanced ability to capture intricate relational patterns, further validating its effectiveness in knowledge graph completion tasks.

5. Conclusion and future work

This study introduces TP-RotatE, a novel knowledge graph (KG) representation learning model that represents entities as complex vectors and relations as rotations within the complex vector space. The model seamlessly integrates path information and contextual rules to address complex relation types, enabling a more holistic representation of KG relationships. Within the model, rules of lengths 1 and 2 are extracted using Ruleformer by consolidating context information around the head entity. These rules are then incorporated into the embedding framework, alongside relationship path information, to resolve additional relationship combinations. The integration of these rules enhances the generation of relational paths, thereby improving the precision of reasoning. The model adeptly captures hidden relationships by considering a multitude of relationship types. Experimental results indicate that TP-RotatE outperforms all baseline models, validating the effectiveness of employing deep learning for rule extraction and the integration of relationship types and paths to enhance KG prediction accuracy.

5.1. Limitations

Despite its promising performance, TP-RotatE has several limitations: (1) Local Context Dependence: The reasoning process primarily focuses on local contextual information. However, more complex reasoning tasks may require a broader global perspective to provide richer semantic understanding. Enhancing the model's global perception could improve its ability to handle more intricate inference tasks. (2) Long-Path Information Representation: The current model may struggle to effectively learn from longer relational paths, leading to under-representation of global knowledge dependencies. This limitation impacts the model's ability to capture deeper relational structures in knowledge graphs.

5.2. Future work

To address these limitations, future research will focus on the following directions: (1) Enhancing Global Awareness: We aim to explore strategies for integrating global knowledge representations to improve semantic understanding in complex reasoning tasks. This includes developing hybrid architectures that combine local path-based inference with global KG structural information. (2) Scalability Optimization: Future work will extend TP-RotatE to large-scale KGs, optimizing computational efficiency to reduce resource consumption. Techniques such as model pruning, efficient sampling strategies, and distributed learning will be explored to enhance scalability. (3) Improving Long-Path Representation: More advanced sequence modeling techniques will be introduced to handle longer and more complex paths, ensuring better knowledge propagation across distant entities. Potential solutions include leveraging transformer-based architectures or recurrent relational reasoning mechanisms.

By addressing these limitations, we aim to further enhance the applicability and robustness of TP-RotatE in real-world KG reasoning tasks.

Acknowledgments

Thanks for all the help of the teachers and students of the related universities.

Author contributions

Conceptualization: Xinliang Liu.

Data curation: Yushi Xu.

Formal analysis: Xinliang Liu.

Funding acquisition: Xinliang Liu.

Methodology: Yanyan Shi.

Software: Yanyan Shi.

Validation: Yanzhao Ren.

Writing – original draft: Yanyan Shi.

Writing – review & editing: Yanzhao Ren.

References

1. Ehrlinger L, Wöß W. Towards a Definition of Knowledge Graphs. 2016 [cited 2024 Dec 17]. Available from: <https://www.semanticscholar.org/paper/Towards-a-Definition-of-Knowledge-Graphs-Ehrlinger-W%C3%B6ß%C3%9F/b18e4272a7b9fa2e1c970d258ab5ea99ed5e2284>
2. Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data [Internet]. New York, NY, USA: Association for Computing Machinery; 2008 [cited 2024 Oct 23]. p. 1247–50. (SIGMOD '08). Available from: <https://dl.acm.org/doi/10.1145/1376616.1376746>
3. Dessì D, Osborne F, Reforgiato Recupero D, Buscaldi D, Motta E, Sack H. AI-KG: An Automatically Generated Knowledge Graph of Artificial Intelligence. In: Pan JZ, Tamia V, d'Amato C, Janowicz K, Fu B, Polleres A, et al., editors. The Semantic Web – ISWC 2020. Cham: Springer International Publishing; 2020. p. 127–43.
4. Zhao L, Xu Y, Wang Y, Chen Z, Huang Y, Feng Q. RPR-KGQA: Relational Path Reasoning for Multi-hop Question Answering with Knowledge Graph. In: Proceedings of the 2024 International Conference on Computer and Multimedia Technology [Internet]. Sanming China: ACM; 2024 [cited 2024 Oct 24]. p. 596–600. Available from: <https://dl.acm.org/doi/10.1145/3675249.3675353>
5. Wang M, Li Z, Wang J, Zou W, Zhou J, Gan J. TracKGE: Transformer with Relation-pattern Adaptive Contrastive Learning for Knowledge Graph Embedding. Knowledge-Based Systems. 2024;301:112218. <https://doi.org/10.1016/j.knosys.2024.112218>
6. Nie W, Bao Y, Zhao Y, Liu A. Long Dialogue Emotion Detection Based on Commonsense Knowledge Graph Guidance. IEEE Trans Multimedia. 2024;26:514–28. <https://doi.org/10.1109/tmm.2023.3267295>
7. Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining [Internet]. New York New York USA: ACM; 2014. p. 601–10. Available from: <https://dl.acm.org/doi/10.1145/2623330.2623623>
8. Tian J, Wang W, Wang Z, Qian Z, Zhang W. EVADE: targeted adversarial false data injection attacks for state estimation in smart grid. IEEE Trans Sustain Comput. 2024;29(4):1234–45. <https://doi.org/10.1109/TSUSC.2023.3278945>
9. Sellami D, Inoubli W, Farah IR, Aridhi S. Knowledge graph representation learning: A comprehensive and experimental overview. Computer Science Review. 2025;56:100716. <https://doi.org/10.1016/j.cosrev.2024.100716>
10. Bordes A, Usunier N, García-Durán A, Weston J, Yakhnenko O. Translating Embeddings for Modeling Multi-relational Data. 2013 [cited 2024 Oct 24]. Available from: <https://www.semanticscholar.org/paper/Translating-Embeddings-for-Modeling-Data-Bordes-Usunier/2582ab7c70c9e7fc-b84545944eba8f3a7f253248#paper-topics>
11. Wang Z, Zhang J, Feng J, Chen Z. Knowledge Graph Embedding by Translating on Hyperplanes. AAAI [Internet]. 2014 Jun 21 [cited 2024 Oct 24];28(1). Available from: <https://ojs.aaai.org/index.php/AAAI/article/view/8870>
12. Seo S, Oh B, Lee KH. Reliable knowledge graph path representation learning. IEEE Access. 2020;8:32816–25.
13. Lin X, Liang Y, Giunchiglia F, Feng X, Guan R. Relation path embedding in knowledge graphs. Neural Comput & Applic. 2018;31(9):5629–39. <https://doi.org/10.1007/s00521-018-3384-6>
14. Niu G, Zhang Y, Li B, Cui P, Liu S, Li J, et al. Rule-Guided Compositional Representation Learning on Knowledge Graphs. AAAI. 2020 Apr 3;34(03):2950–8.
15. Niu G, Li B, Zhang Y, Sheng Y, Shi C, Li J, et al. Joint semantics and data-driven path representation for knowledge graph reasoning. Neurocomputing. 2022;483:249–61. <https://doi.org/10.1016/j.neucom.2022.02.011>
16. Yang B, Yih WT, He X, Gao J, Deng L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. International Conference on Learning Representations. 2014.
17. Trouillon T, Welbl J, Riedel S, Gaussier E, Bouchard G. Complex Embeddings for Simple Link Prediction. In: Proceedings of The 33rd International Conference on Machine Learning [Internet]. PMLR; 2016 [cited 2024 Oct 24]. p. 2071–80. Available from: <https://proceedings.mlr.press/v48/trouillon16.html>
18. Wang Y, Xiao W, Tan Z, Zhao X. Caps-OWKG: a capsule network model for open-world knowledge graph. Int J Mach Learn & Cyber. 2021;12(6):1627–37. <https://doi.org/10.1007/s13042-020-01259-4>

19. Lin Y, et al. Modeling Relation Paths for Representation Learning of Knowledge Bases. EMNLP. 2015;705–14.
20. Jin L, Yao Z, Chen M, Chen H, Zhang W. A Comprehensive Study on Knowledge Graph Embedding over Relational Patterns Based on Rule Learning. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 14265 LNCS. 2023.p. 290–308. https://doi.org/10.1007/978-3-031-47240-4_16
21. Dong H, Ning Z, Wang P, Qiao Z, Wang P, Zhou Y, Fu Y. Adaptive Path-Memory Network for Temporal Knowledge Graph Reasoning. IJCAI International Joint Conference on Artificial Intelligence, 2023-August, 2023. p. 2086–94. <https://doi.org/10.24963/ijcai.2023/232>
22. Feng J, Huang M, Wang M, Zhou M, Hao Y, Zhu X. Knowledge Graph Embedding by Flexible Translation. In Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning.
23. AMIE | Proceedings of the 22nd international conference on World Wide Web [Internet]. [cited 2024 Oct 24]. Available from: <https://dl.acm.org/doi/abs/10.1145/2488388.2488425>
24. Yu D, Xu G, Jing Z, Li X. Improving Representation Learning Incorporating Neighborhood Context and Relational Paths for Knowledge Graph Completion. In: 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService) [Internet]. 2021 [cited 2024 Oct 24]. p. 161–8. Available from: <https://ieeexplore.ieee.org/abstract/document/9564341>
25. Galárraga L, Teflioudi C, Hose K, Suchanek F. Fast rule mining in ontological knowledge bases with AMIE. VLDB J. 2015;24(6):707–30.
26. Galárraga LA, Teflioudi C, Hose K, Suchanek F. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the 22nd international conference on World Wide Web [Internet]. Rio de Janeiro Brazil: ACM; 2013 [cited 2024 Oct 24]. p. 413–22. Available from: <https://dl.acm.org/doi/10.1145/2488388.2488425>
27. Meilicke C, Fink M, Wang Y, Ruffinelli D, Gemulla R, Stuckenschmidt H. Fine-Grained Evaluation of Rule- and Embedding-Based Systems for Knowledge Graph Completion. In: Vrandečić D, Bontcheva K, Suárez-Figueroa MC, Presutti V, Celino I, Sabou M, et al., editors. The Semantic Web – ISWC 2018. Cham: Springer International Publishing; 2018. p. 3–20.
28. Saeedan F, Weber N, Goesele M, Roth S. Detail-Preserving Pooling in Deep Networks. 2018 [cited 2024 Oct 24]. p. 9108–16. Available from: https://openaccess.thecvf.com/content_cvpr_2018/html/Saeedan_Detail-Preserving_Pooling_in_CVPR_2018_paper.html
29. Cohen WW. TensorLog: A Differentiable Deductive Database [Internet]. arXiv; 2016 [cited 2024 Oct 24]. Available from: <https://arxiv.org/abs/1605.06523>
30. Li J, Wang R, Zhang N, Zhang W, Yang F, Chen H. Logic-guided Semantic Representation Learning for Zero-Shot Relation Classification. Proceedings of the 28th International Conference on Computational Linguistics. 2020.
31. Wang P, Stepanova D, Domokos C, Kolter Z. Differentiable learning of numerical rules in knowledge graphs. ICLR.
32. Yang F, Yang Z, Cohen WW. Differentiable Learning of Logical Rules for Knowledge Base Reasoning | Semantic Scholar [Internet]. [cited 2024 Oct 24]. Available from: <https://www.semanticscholar.org/paper/Differentiable-Learning-of-Logical-Rules-for-Base-Yang-Yang/23e42bc79f10234bdceef-31441be39a2d9d2a9a0>
33. Sadeghian A, Armandpour M, Ding P, Wang DZ. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs [Internet]. arXiv; 2019. [cited 2024 Oct 24]. Available from: <http://arxiv.org/abs/1911.00055>