

Article Relgraph: A Multi-Relational Graph Neural Network Framework for Knowledge Graph Reasoning Based on Relation Graph

Xin Tian ^{1,*} and Yuan Meng ^{2,*}



- ² Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
- * Correspondence: tianxin07@nudt.edu.cn (X.T.); yuanmeng@mail.tsinghua.edu.cn (Y.M.)

Abstract: Multi-relational graph neural networks (GNNs) have found widespread application in tasks involving enhancing knowledge representation and knowledge graph (KG) reasoning. However, existing multi-relational GNNs still face limitations in modeling the exchange of information between predicates. To address these challenges, we introduce Relgraph, a novel KG reasoning framework. This framework introduces relation graphs to explicitly model the interactions between different relations, enabling more comprehensive and accurate handling of representation learning and reasoning tasks on KGs. Furthermore, we design a machine learning algorithm based on the attention mechanism to simultaneously optimize the original graph and its corresponding relation graph. Benchmark and experimental results on large-scale KGs demonstrate that the Relgraph framework improves KG reasoning performance. The framework exhibits a certain degree of versatility and can be seamlessly integrated with various traditional translation models.

Keywords: knowledge graph; reasoning; multi-relational; predicate



Citation: Tian, X.; Meng, Y. Relgraph: A Multi-Relational Graph Neural Network Framework for Knowledge Graph Reasoning Based on Relation Graph. *Appl. Sci.* 2024, *14*, 3122. https://doi.org/10.3390/ app14073122

Academic Editors: Artur Jorge Ferreira and Johannes Zenkert

Received: 23 February 2024 Revised: 2 April 2024 Accepted: 2 April 2024 Published: 8 April 2024



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

1. Introduction

In the field of Artificial Intelligence, knowledge graph (KG) reasoning has become a pivotal research topic. As a powerful representation, KG integrates billions of available relational facts. Its importance lies in its potential to enhance the representation and inference of knowledge, enabling more intelligent decision making and problem solving.

KG is now utilized in various downstream applications, including recommendation systems [1], query answering [2], and drug discovery [3]. However, due to constraints in human knowledge and text extraction technology, even the largest KGs remain incomplete. Given that the construction of knowledge graphs involves extensive data and complex semantic relationships, there are inevitable omissions and defects. To address these issues, knowledge graph reasoning techniques fill in the gaps and optimize the knowledge graph, with a typical task being knowledge graph completion (KGC). Currently, KG reasoning has become an important research field, attracting the attention and exploration of numerous researchers. With the continuous advancement of technology and increasing application demands, KG reasoning will play a more significant role in future intelligent applications.

Multi-relational graph neural networks (GNNs) [4,5] extend traditional GNNs by considering multiple relations in the graph as distinct node attributes. This extension enables the model to capture a broader range of semantic information and better encode the contextual relationships between entities. By incorporating multiple relations, multi-relational GNNs are able to handle more complex reasoning tasks, such as transitive reasoning, role reversal, and collective entity recognition. This increased expressivity significantly improves the accuracy and scalability of knowledge graph reasoning.

Despite their advantages, current multi-relational GNNs face limitations. Firstly, most existing models assume that different relations have independent impacts on entities, ignoring potential interactions between different relations. This limitation can lead to

incomplete or inaccurate reasoning results. Secondly, existing models often treat all relations equally, disregarding their different levels of importance in different reasoning tasks. This uniform treatment may result in information loss or overfitting in complex reasoning tasks. Figure 1 offers an example of the interactions between predicates.



Figure 1. The interaction between predicates of a toy knowledge graph (KG) about kinship relationships. The blue lines and blue squares represent the relations and entities in the KG, respectively. Entity D can be used to analyze the information connections between the two predicates of father and uncle, while entity A can help to analyze the interaction between grandson and father, which are depicted in red lines in the figure.

In this paper, we aim to address the limitations of ignoring interactive effects between relations in existing multi-relational GNNs. We introduce the Relgraph, a novel knowledge graph reasoning framework that explores logical relationships between relations by introducing a *relation graph*. This relation graph serves as a dual graph of the original KG, treating relations of the original KG (or predicates) as entities and entities in the original KG as relations. The Relgraph explicitly models the interaction between different relations as relations on the relation graph (referred to as *entity-links*), enabling more comprehensive and accurate reasoning (corresponding to it, the relations in the original KG are actually *relation-links*). By assigning different weights to different entity links based on their importance in specific reasoning tasks, Relgraph effectively handles complex reasoning tasks while reducing information loss and overfitting. A typical original knowledge graph and its corresponding relation graph are shown in Figure 2.

We extend the traditional graph attention network (GAT) to its dual neural network, the relation graph attention network (RGAT), and design a machine learning algorithm based on the attention mechanism to synchronously optimize the two networks. As a result, we successfully apply the Relgraph framework to knowledge graph reasoning tasks. The Relgraph framework can embed various algorithms for representation learning, such as TransE and RotateE, to optimize the representation learning of entities and relations, ultimately enhancing the reasoning performance on KGs.

Through benchmark testing and conducting a drug repurposing task based on biochemical KGs, we experimentally demonstrate that Relgraph improves the performance of reasoning models on KGs.

The primary contributions of this article are the following.

- We introduce the concept of the relation graph, which extends the capabilities of multi-relational GNNs for knowledge graph reasoning tasks. From this foundation, we present the Relgraph, a novel knowledge graph reasoning framework.
- We design and implement a machine learning mechanism based on GAT that simultaneously optimizes GAT and RGAT using an attention mechanism. This integration allows the Relgraph framework to learn representations from KGs and apply them to reasoning tasks, thereby enhancing the performance of multi-relational GNNs in knowledge graph reasoning tasks.
- The proposed Relgraph framework is versatile and can integrate various representation learning algorithms, such as TransE [6] and RotatE [7].



Figure 2. The toy kinship knowledge graph (KG) (**top**) and its corresponding relation graph (**bottom**). In the knowledge graph above, vertices represent entities, and links represent relations (or predicates) in the knowledge graph. In the relation graph, vertices represent relations from the knowledge graph, links represent entity links (red dashed line). The presence of an entity link between vertices is determined by Equation (2).

2. Related Work

The primary objective of this article is to introduce an innovative approach to representation learning on KGs that leverages the attention mechanism of GNNs for KGC reasoning tasks. To achieve this, we conducted a literature review encompassing KG embeddings for KGC, advancements in GNN-based KGC, and models that incorporate multi-relational GNNs. Unlike traditional embedding-based and GNN-based KGC models, the Relgraph framework excels at capturing relational dependencies within GNNs, enabling more effective KG embedding learning and subsequently enhancing the performance of KGC models.

KGC via KG embeddings. The utilization of learnable embeddings for KGs has spawned numerous works aimed at KGC tasks. For instance, references [6–10] describe methods that learn to map KG relations into vector spaces and employ scoring functions for KGC tasks. In contrast, NTN [11] parameterizes each relation using a neural network. Paper [12] proposes a divide–search–combine algorithm, RelEns-DSC, to efficiently search relation-aware ensemble weights for KG embedding. The common drawback of these methods is that they cannot explicitly model the relationships between predicates.

KGC via GNNs. GNN is a framework introduced by [13] for learning deep models or embeddings on graph-structured data. The theoretical foundation for GNNs' ability to capture common graph heuristics for KGC tasks in simple graphs was established by [14]. Furthermore, ref. [15] employed a similar approach to achieve competitive results on inductive matrix completion. Homogeneous GNN models are often unable to directly apply to tasks involving multiple relations.

Multi-relational GNNs. Multi-relational GNNs have been widely studied in recent years as a powerful tool for enhancing knowledge representation and reasoning on KGs [4,5]. These networks aim to capture the complex relationships between entities and

predicates in KGs. One line of research focuses on introducing more complex network architectures to capture the multi-relational interactions better [16]. These models often involve the use of attention mechanisms or graph convolutional layers to capture the interactions between different relations explicitly. Another line of research aims to effectively develop learning algorithms that can optimize the representations learned by multi-relational GNNs [4]. Paper [17] proposes an adaptive propagation path learning method for GNNbased inductive KG reasoning, addressing the limitations of hand-designed paths and the explosive growth of entities. Relphormer [18] leverages Triple2Seq to handle heterogeneity and a structure-enhanced self-attention mechanism to encode relational information. These algorithms often involve the use of contrastive learning or other optimization techniques to improve the performance of representation learning and reasoning tasks on KGs. However, they often cannot explicitly model the relationships between predicates, which limits their accuracy and efficiency in conducting knowledge graph reasoning.

3. Materials and Methods

3.1. Preliminaries

KGs. The KG is a structured representation of facts in the form of triplet information. It is defined as a set of triples \mathcal{K} containing elements (h, r, t) where h and t belong to the set of entities \mathcal{E} , and r belongs to the set of predicates \mathcal{R} . A triple $(h, r, t) \in \mathcal{K}$ represents a fact in the KG, where h and t are entities and r is a relation, $\mathcal{K} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$.

To avoid confusion, we use the terms "relation" and "predicate" interchangeably throughout this discussion, as they carry the same meaning in our context.

When referring to the triple (h, r, t), h and t are called the "head entity" and "tail entity", respectively. The sets of all head entities and tail entities corresponding to a particular predicate r are denoted as \mathcal{HD}_r and \mathcal{TL}_r , respectively.

3.2. The Definition of Relgraph

Meta-nodes. In this paper, we introduce the concept of *meta-nodes* to represent the relationship between entities and predicates in KGs. These meta-nodes are divided into two categories: entity meta-nodes and predicate meta-nodes. Entity meta-nodes, denoted as *emn_{i,a}*, represent the component of entity e_i corresponding to predicate r_a . Entity meta-nodes express the influence of entities on predicates. The set \mathcal{EMN} represents all such meta-nodes in the KG, formally defined as $\{emn_{i,a} | e_i \in \mathcal{E}, r_a \in \mathcal{R}\}$, and belongs to the cross product $\mathcal{E} \times \mathcal{R}$. Similarly, \mathcal{EMN}_a represents all entity meta-nodes associated with a specific predicate r_a .

For the relation $r_a \in \mathcal{R}$, we define the graph $\mathcal{G}_a = \langle \mathcal{V}_a, r_a, \mathcal{D}_a \rangle$, where the set of edges \mathcal{D}_a contains triples $(h, r_a, t), h, t \in \mathcal{V}_a$ represent source and target vertices, respectively, and r_a is the unique relation of the graph \mathcal{G}_a . For each fact $(e_m, r_a, e_n) \in \mathcal{K}$, let h, t represent the entity meta-nodes $emn_{m,a}, emn_{n,a} \in \mathcal{EMN}_a$, respectively. Then, the triple (h, r_a, t) corresponds to the fact (e_m, r_a, e_n) in the knowledge graph. For a triple $(emn_{m,a}, r_a, emn_{n,a}) \in \mathcal{D}_a$ associated with entities $e_m, e_n \in \mathcal{E}$ related to the predicate $r_a \in \mathcal{R}$, we define the logical function:

$$\Gamma F(emn_{m,a}, r_a, emn_{n,a}) = \begin{cases} True & (e_m, r_a, e_n) \in \mathcal{K}, \\ False & (e_m, r_a, e_n) \notin \mathcal{K}. \end{cases}$$
(1)

Equation (1) is referred as the truth function of graph G_a .

Dually, for $e_i \in \mathcal{E}$ and $r_a \in \mathcal{R}$, we set $pmn_{i,a}$ as the predicate meta-node representing the influence of predicate r_a on entity e_i . The set $\mathcal{PMN} = \{pmn_{i,a} | e_i \in \mathcal{E}, r_a \in \mathcal{R}\} \subseteq \mathcal{R} \times \mathcal{E}$ represents all predicate meta-nodes in the knowledge graph. Similarly, $\mathcal{PMN}_i = \{pmn_{i,a} | r_a \in \mathcal{R}\}$ represents all predicate meta-nodes associated with entity e_i .

Entity Links. Let $e_i \in \mathcal{E}$ be an entity. We define a graph $\mathcal{RG}_i = \langle \mathcal{RV}_i, el_i, \mathcal{RD}_i \rangle$ such that each edge $rd \in \mathcal{RD}_i$ is a triple (h, el_i, t) where $h, t \in \mathcal{RV}_i$ are source and destination vertices. The only relation in \mathcal{RG}_i is el_i . For predicates $r_f, r_g \in \mathcal{R}$, let h, t be predicate meta-nodes $pmn_{i,f}, pmn_{i,g} \in \mathcal{PMN}_i$, define el_i as an *entity link*. The graph \mathcal{RG}_i is called a

relation graph about entity link el_i . To distinguish it from entity graphs, we refer to \mathcal{G}_a as an *entity graph*.

It follows from the definition that for each entity $e_i \in \mathcal{E}$, there exists a corresponding collection of predicate meta-nodes \mathcal{PMN}_i and an entity link el_i . The collection of all entity links is denoted by \mathcal{EL} , and $|\mathcal{EL}| = |\mathcal{E}|$. To reflect the influence of entities on predicate pairs, we extend the truth function Equation (1) to relation graphs. The truth function for relation graph \mathcal{RG}_i is defined as follows:

$$TF(pmn_{i,f}, el_i, pmn_{i,g}) = \begin{cases} True & \exists e_x, e_y \in \mathcal{E}, \\ (e_x, r_f, e_i) \in \mathcal{K} \\ and \\ (e_i, r_g, e_y), \in \mathcal{K}, \end{cases}$$
(2)
False else.

The visual representation of the Truth function for relation graphs can be found in Figure 2. It shows that if an entity e_i is the tail entity of a fact (e_x, r_f, e_i) and the head entity of a fact (e_i, r_g, e_y) , and the predicate meta-nodes $pmn_{i,f}, pmn_{i,g}$ represent the influence of predicate r_f, r_g on e_i , respectively, then $P(pmn_{i,f}, el_i, pmn_{i,g}) = True$. In analogy to the definition of facts in knowledge graphs, for any $pmn_{i,f}, pmn_{i,g} \in \mathcal{PMN}_i, el_i \in \mathcal{EL}$, if $P(pmn_{i,f}, el_i, pmn_{i,g}) = True$, we define (r_f, el_i, r_g) as a *relation-fact*. The collection of all relation-facts is denoted by $\mathcal{RK} = \{(h, el, t) | h, t \in \mathcal{R}, el \in \mathcal{EL}\} \subseteq \mathcal{R} \times \mathcal{E} \times \mathcal{R}$.

3.3. Relgraph Attention Network

Knowledge graph embedding (KGE) represents entities and relations in a continuous space called *embeddings*. A scoring function based on these embeddings can be defined to score triplets $(h, r, t), h, t \in \mathcal{E}, r \in \mathcal{R}$. The embeddings are trained to ensure that observed facts in the KG have higher scores than unobserved ones. Well-trained embeddings are typically used for tasks like KGC and rule mining.

According to the definition provided in Section 4, we have transformed the knowledge graph into $|\mathcal{R}|$ entity graphs $\{\mathcal{G}_a | r_a \in \mathcal{R}\}$, each containing $|\mathcal{E}|$ vertices representing entity meta-nodes. Additionally, we have $|\mathcal{E}|$ relation graphs $\{\mathcal{RG}_i | e_i \in \mathcal{E}\}$, each containing $|\mathcal{R}|$ vertices representing predicate meta-nodes. In the entity graph, the relations are defined by the predicates present in the KG, while in the relation graph, the relations are defined by entity-links. To perform tasks, such as KGC, Relgraph utilizes two interconnected graph attention networks to train embeddings for entities, predicates, and entity links. These networks are referred to as the *Entity GAT* (graph attention network) and the *Relation GAT* (relation graph attention network). The two GATs share the same embedding of predicates.

Entity GAT. The Entity GAT corresponding to entity graph \mathcal{G}_k is responsible for analyzing the message passing process of Relgraph and the attention between vertices $emn_{i,k}$ and $emn_{j,k}$ (entity meta-nodes representing the influence of entity e_i , e_j on the predicate r_k , respectively). We denote e_{ijk} as the component of $emn_{i,k}$ corresponding to the vertex $emn_{j,k}$. Matrix $F \in \mathbb{R}^{|\mathcal{E}| \times d}$ represents the input features, where d is the dimension of the feature. Each row $f_i = F_i$: represents the embedding of entity e_i . Similarly, matrix $P \in \mathbb{R}^{|\mathcal{R}| \times d}$ represents features of the predicates, and each row $p_k = P_k$: represents the embedding of predicate r_k . We denote f_{ijk} as the embedding of e_{ijk} .

In accordance with [16], the entity GAT uses the following aggregation function to update the embedding of entities and predicates:

$$f_{ijk} = W_f[f_i|f_j|p_k],\tag{3}$$

$$b_{ijk} = ReLu(W_b f_{ijk}), \tag{4}$$

$$a_{ijk} = \frac{exp(b_{ijk})}{\sum_{n \in N_i} \sum_{r \in R_{in}} exp(b_{inr})},$$
(5)

$$\hat{f}_i = \sigma(\sum_{j \in N_i} \sum_{k \in R_{ij}} a_{ijk} f_{ijk}),$$
(6)

$$F' = W_F F + \hat{F},\tag{7}$$

$$P' = W_P P, \tag{8}$$

where p_k is the embedding of predicate r_k determined by $(e_j, r_k, e_i) \in \mathcal{K}$, neighbor degree $N_i = \{j | e_j \in \mathcal{E}, r_k \in \mathcal{R}, (e_j, r_k, e_i) \in \mathcal{K}\}$ is the number set of e_i 's one-hop neighboring vertices, and N_i also includes i (i.e., there is a self-loop); $R_{ij} = \{k | e_i \in \mathcal{E}, e_j \in \mathcal{E}, (e_j, r_k, e_i) \in \mathcal{K}\}$ is the number set of predicates that have appeared in all facts with e_i as the head entity and e_j as the tail entity; a_{ijk} is the attention weight between the target vertex $emn_{i,k}$ and the neighboring vertex $emn_{j,k}$, which is generated by applying softmax to the values computed by b_{ijk} . The parameters W_f , W_b , W_F , and W_P are trainable parameters of the attention function.

Relation GAT. Dually, for each relation graph \mathcal{RG}_s in the Relgraph, to comprehend the message-passing process and the attention between vertices pmn_{us} and pmn_{vs} (predicate meta-nodes that symbolize the impact of predicates r_u, r_v on entity e_s , respectively), we utilize r_{uvs} as the component of pmn_{us} at vertex pmn_{vs} . Matrix $Q \in \mathbb{R}^{|\mathcal{E}| \times d}$ represents the features of entity links, with each row $q_e = Q_e$: denoting the embedding of entity link el_e . We also denote p_{uvs} as the embedding of r_{uvs} .

After updating the embedding of entities and predicates using Equations (7) and (8) via entity GAT, Relation GAT employs the aggregation function below to update the embedding of entity links and predicates once again.

$$p_{uvs} = W_p[p_u|p_v|q_s], \tag{9}$$

$$d_{uvs} = ReLu(W_d p_{uvs}), \tag{10}$$

$$b_{uvs} = \frac{exp(d_{uvs})}{\sum_{n \in QN_u} \sum_{x \in EL_{un}} exp(d_{unx})},$$
(11)

$$\hat{p_u} = \sigma(\sum_{v \in QN_u} \sum_{s \in EL_{uv}} b_{uvs} d_{uvs}),$$
(12)

$$P' = W_{PQ}P + \hat{P},\tag{13}$$

$$Q' = W_Q Q, \tag{14}$$

where q_s represents the embedding of entity link el_s according to the relation-fact $(r_v, el_s, r_u) \in \mathcal{RK}$, $QN_u = \{v | r_v \in \mathcal{R}, el_s \in \mathcal{EL}, (r_v, el_s, r_u) \in \mathcal{RK}\}$ is the number set of r_u 's one-hop neighboring vertices, and QN_u also includes u (i.e., there is a self-loop on each vertex); $EL_{uv} = \{s | r_u \in \mathcal{R}, r_v \in \mathcal{R}, (r_v, el_s, r_u) \in \mathcal{RK}\}$ is the number set of entity-links that have appeared in all relation-facts with r_v as the head entity and r_u as the tail entity; b_{uvs} is the attention weight between the target vertex $pmn_{u,s}$ and the neighboring vertex $pmn_{v,s}$, which is computed by applying softmax to the values determined by d_{uvs} . W_p , W_d , W_{PQ} , and W_Q are the trainable parameters of the attention function.

3.4. Training Algorithm

We train Relgraph through KGC tasks on KGs for embedding learning. For entity GAT, we utilize the training dataset $\mathcal{K}_{train} \subset \mathcal{K}$, as well as the negative sample training set \mathcal{K}'_{train} that is generated from \mathcal{K}_{train} in a 3:1 ratio, where

$$\mathcal{K}'_{train} = \{ (h', r, t') | (h, r, t) \in \mathcal{K}_{train}, \\ (h', r, t') \notin \mathcal{K}_{train}, h' \neq h \lor t' \neq t \}.$$

$$(15)$$

For Relation GAT, the model generates the relation facts as training set $\mathcal{RK}_{train} \subset \mathcal{RK}$ based on \mathcal{K}_{train} , using Equation (2). Similarly, the negative sample training set \mathcal{RK}'_{train} is generated following the same method.

The scoring function $SC(e_i, r_a, e_j)$ predicts the probability of $(e_i, r_a, e_j) \in \mathcal{K}$ being true. For relation facts $(r_a, el_j, r_b) \in \mathcal{RK}$, the model employs the same scoring function $SC(r_a, el_j, r_b)$. In each iteration, the model minimizes the following loss:

$$\mathcal{L}_{K} = \sum_{(h,r,t) \in \mathcal{K}_{train}} \sum_{(h',r,t') \notin \mathcal{K}'_{train}} [\max(0, SC(h',r,t') - SC(h,r,t) + \gamma)],$$
(16)

$$\mathcal{L}_{RK} = \sum_{(h,r,t) \in \mathcal{RK}_{train}} \sum_{(h',r,t') \notin \mathcal{RK}'_{train}}$$
(17)

$$[\max(0, SC(h', r, t') - SC(h, r, t) + \gamma)],$$

$$\mathcal{L}_{RK} = \mathcal{L}_K + \eta \mathcal{L}_{RK},\tag{18}$$

where γ denotes the margin hyperparameter and η denotes the RGAT weight hyperparameter used to control the penalty for incorrect predictions in the relation graph.

3.5. Variants of Relgraph

The Relgraph framework can be seamlessly integrated with TransE, RotatE, and other transitive representation learning methods by utilizing distinct scoring functions. Based on the scoring function utilized, Relgraph can be classified into distinct variants. We have developed the Relgraph-RotatE and Relgraph-TransE versions of the model, along with the GAT-RotatE version for comparative experiments that do not involve entity links.

 Relgraph-TransE. In the Relgraph-TransE version, we use the following formula as the scoring function:

$$SC(h,r,t) = |f_h + p_r - f_t|, \forall (h,r,t) \in \mathcal{K} \cup \mathcal{RK}.$$
(19)

Here, f_h , f_t represent the embeddings of h, t, respectively, and p_r represents the embedding of r.

 Relgraph-RotatE. In the original RotatE model, entity embeddings consist of both the real and imaginary parts, which are operated separately with the predicate embeddings representing the rotation angle in the complex space. Consequently, the dimensionality of entity embeddings needs to be twice that of the predicates. However, in Relgraph, there exists a duality between entities and predicates, requiring the dimensions of entities and predicates to be consistent. This contradiction necessitates a special design for Relgraph-RotatE.

Let p_r be the embedding of a predicate or entity link r with a dimensionality of D. We can represent p_r as the concatenation of two parts: p_r^u and p_r^d . Here, p_r^u represents the first $\frac{D}{2}$ dimensions of p_r , and p_r^d represents the last $\frac{D}{2}$ dimensions of p_r . Mathematically, we can express it as $p_r = Cat(p_r^u, p_r^d)$, where *Cat* represents the concatenation operation of vectors. In Relgraph-RotatE, the rotation angle from the head vertex to the tail vertex in a triplet is calculated using the following equation:

$$p_r^{\theta} = \min(p_r^u + p_r^d, p_r^u - p_r^d),$$
(20)

where p_r^{θ} represents the measurement of the rotation angle from the head entity to the tail entity in RotatE. The scoring function used in Relgraph-RotatE is as follows:

$$SC(h,r,t) = |f_h \circ p_r^{\theta} - f_t|, \forall (h,r,t) \in \mathcal{K} \cup \mathcal{RK},$$
(21)

where f_h , f_t denotes the embeddings of h, t. The reason for this design is due to the discontinuous nature of the following equation (Equation (23)):

$$p_{r}^{\theta} = \begin{cases} p_{r}^{u} + p_{r}^{d} & p_{r}^{u} < 0 \land p_{r}^{d} < 0, \\ p_{r}^{u} - p_{r}^{d} & else. \end{cases}$$
(22)

This discontinuous nature allows the representation learning model to better distinguish between entities and predicates that result in true or false triple assignments. It also helps to maintain a relatively clear and stable geometric interpretation of the dimensions in the embeddings.

 GAT-RotatE and GAT-TransE. We employ GAT-RotatE and GAT-TransE as benchmark models to assess the impact of introducing entity links on model performance. By setting η in Equation (18) to 0, we obtain GAT-RotatE/GAT-TransE from Relgraph-RotatE/Relgraph-TransE.

4. Results

We have conducted rigorous experiments to demonstrate the efficacy of Relgraph in enhancing the performance of reasoning models on KGs. These experiments include KGC experiments on the benchmark dataset as well as an experiment about drug repurposing.

4.1. Experimental Setup

Benchmark datasets. The evaluation of open-world knowledge KGC tasks often relies on subsets of Word-Net and Freebase, such as WN18RR [10] and FB15K-237 [19]. To verify the effectiveness of Relgraph, we need to choose datasets with numerous predicates and high difficulty levels to validate our method's efficacy. Therefore, we have selected FB15K-237, WN18RR, and UMLS [20] as our experimental datasets. UMLS is a domain-specific knowledge graph in the medical domain, containing biomedical concepts and their relationships.

Drug repurposing datasets. To verify the efficacy of this method for knowledge extraction and logical reasoning on large-scale datasets, we conducted drug repurposing experiments on the open-source biochemical knowledge graph RTX-KG2c [21]. RTX-KG2c integrates data from 70 public knowledge sources into a comprehensive graph where all biological entities (e.g., "ibuprofen") are represented as nodes and all concept–predicate–concept relationships (e.g., "ibuprofen–increased activity–GP1BA gene") are encoded as edges. This dataset comprises approximately 6.4 M entities across 56 distinct categories, with 39.3 M relationship edges described by 77 distinct relations.

Drug repurposing, also known as drug rediscovery or drug repositioning, refers to discovering a new indication for an existing medication. The objective of this experiment is to employ the KGC model to learn the interactions between diseases and drugs from RTX-KG2c, aiming to predict potential therapeutic relationships between drugs and diseases. To identify "new" applications for existing drugs, therapeutic relationships were retrieved from external databases, including MyChem [22] and SemMedDB datasets [23].

To prevent information leakage during training, we excluded all existing edges connecting potential drug nodes (nodes labeled "Drug" or "SmallMolecule") with potential disease nodes (nodes labeled "Disease", "PhenotypicFeature", "BehavioralFeature," or "DiseaseOrPhenotypicFeature") in RTX-KG2c. We then added drug–disease pairs that were confirmed true positives (pairs with the relation "indication" from MyChem Datasets or the predicate "treats" from SemMedDB Datasets). A new predicate treat was introduced to represent this therapeutic relationship in the experimental KG.

We generated new triples based on these drug–disease pairs and added them to the KG, dividing them into training, validation, and testing sets in a 7:2:1 ratio.

Baselines. To test the effectiveness and versatility of Relgraph, we conducted an extensive selection of well-established knowledge graph reasoning models. We used these models on the benchmarks as the baseline. For each baseline, we used them in conjunction with Relgraph to conduct performance testing and record their performance

improvement. The chosen baseline models include: TransE, DistMult [9], RotatE and ConvE [10], CompGCN [4], etc.

In the drug repurposing experiments, all models were trained using the modified RTX-KG2c training dataset. However, when tested on the test set, only the predicted results of triples associated with the predicate treat were considered.

Experiment setting details. We set the entity and relation embedding dimensions to 200 for our experiments. To optimize the model, we utilized the Adam optimization algorithm [24]. We experimented with different learning rates within the range of 0.002 to 0.006, as well as mini-batch sizes from 64 to 256. Additionally, we applied dropout regularization to both the entity and relation embeddings, as well as all feed-forward layers. We searched for an optimal dropout rate within the range of 0.55.

In line with the common practices mentioned in [25,26], we utilize standard evaluation metrics for the link prediction task: Hit@1, which represents the number of correctly predicted head terms among the top 1 predictions, and mean reciprocal ranking (MRR), calculated as the mean of the reciprocal rank of the correct answer in the list of predictions.

To generate predictions, we feed the predicate and entity representations learned from each model version into a ConvKB model [27]. We then train this ConvKB model to act as a decoder, assigning scores to candidate head and tail entities based on the scores given by the ConvKB decoder for each triplet in the test set. Finally, we sort these entities based on these scores, calculate the corresponding metrics (Hit@K and MRR), and evaluate the performance of each model.

All experiments were conducted on a machine equipped with 6 Nvidia Tesla V100 GPUs and 32 GB RAM (Beijing, China). We used the PyTorch library in Python for implementation.

Analysis of computational complexity in experiments. Let *d* be the dimension of embeddedings in the model, it can be analyzed that the spatial complexity of Relgraph is $\mathcal{O}(2|\mathcal{E}|d + |\mathcal{R}|d)$ and the temporal complexity is $\mathcal{O}(|\mathcal{K}|d^2 + |\mathcal{K}|d + |\mathcal{R}\mathcal{K}|d^2 + |\mathcal{R}\mathcal{K}|d)$. In contrast, the TransE model has a spatial complexity of $\mathcal{O}(|\mathcal{E}|d + |\mathcal{R}|d)$ and a temporal complexity of $\mathcal{O}(|\mathcal{E}|d^2 + |\mathcal{K}|d)$. Without considering the relation graph, the spatial complexity of Entity GAT (multi-relational GAT) is $\mathcal{O}(|\mathcal{E}|d + |\mathcal{R}|d)$ and the temporal complexity is $\mathcal{O}(|\mathcal{K}|d^2 + |\mathcal{K}|d)$. It can be seen that compared to GAT and multi-relational GAT models, the complexity of Relgraph only increases linearly. In the KGC experiment on the FB15K-237 dataset, when d = 200, Relgraph can complete training in about 1 h.

4.2. Experiment Results

KGC tasks on benchmarks. As shown in Table 1, in the KGC tasks, the performance of Relgraph is optimal on most metrics. Relgraph-TransE and Relgraph-RotatE have better performance than other representation-based methods such as TransE, RotatE, and ConvE. For MRR on FB15K-237 and UMLS database, Relgraph-RotatE is 56.5% and 22.0% higher than the best algorithm before Relgraph and the improvement of Relgraph-TransE is 52.9% and 21.6%, respectively.

Table 1. Results of knowledge graph completion (KGC) tasks on benchmarks. * denotes results from publications. - denotes the unpublished results. The experimental results indicate that the KGC performance of Relgraph-RotatE and Relgraph-TransE exceeds traditional methods.

Algorithm	WN18RR				FB15K-237				UMLS			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE *	0.222	1.3	40	53	0.3	23	36.9	52.7	-	-	-	-
DistMult	0.444	41.2	47	50.4	0.281	19.9	30.1	44.6	0.391	25.6	44.5	66.9
ConvE *	0.456	41.9	47	53.1	0.312	22.5	34.1	49.7	-	-	-	-
CompGCN	0.473	42.6	47.8	55.1	0.337	23.8	35.1	49.9	0.721	58.6	81.1	90.1
RotatE	0.476	42.8	49.2	57.1	0.338	24.1	37.5	53.3	0.744	63.6	82.2	93.9
Ours:Relgraph-TransE	0.478	41.3	51.9	59.9	0.517	45.7	54.4	63.6	0.905	84.4	95.8	98.3
Ours:Relgraph-RotatE	0.485	43.2	52.5	61.5	0.529	46.7	55.6	64.9	0.908	86.6	94.1	97.2

The impact of relation graph on enhancing the performance of reasoning. Table 2 shows the improvement brought by relation graph in embedding learning and KG reasoning. It can be found that the model version using relation graph (Relgraph-RotatE or Relgraph-TransE) has certain advantages in performance compared to the version not used (GAT-RotatE or GAT-TransE).

Table 2. The impact of relation graph on enhancing the performance of embedding learning and knowledge graph (KG) reasoning. * denotes results from publications. The experimental results indicate that the introduction of relation graph can improve the accuracy of KG reasoning.

Methods	WN18RR				FB15K-237				UMLS			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE *	0.222	1.3	40	53	0.3	23	36.9	52.7	_	-	-	-
RotatE	0.476	42.8	49.2	57.1	0.338	24.1	37.5	53.3	0.744	63.6	82.2	93.9
GAT-RotatE	0.472	39.1	51.8	60.7	0.501	44.5	52.6	62.7	0.862	76.5	91.3	96.7
GAT-TransE	0.468	38.5	50.0	56.9	0.488	43.2	51.1	60.7	0.829	73.7	88.3	95.0
Relgraph-RotatE	0.485	43.2	52.5	61.5	0.529	46.7	55.6	64.9	0.908	86.6	94.1	97.2
Relgraph-TransE	0.478	41.3	51.9	59.9	0.517	45.7	54.4	63.6	0.905	84.4	95.8	98.3

We use a bar chart to visually demonstrate the improvement of KG reasoning ability through relation graph in Relgraph, as shown in Figure 3.



Figure 3. MRR improvement brought by relation graph in Relgraph on benchmarks.

Drug repurposing. As shown in Table 3, in the drug repurposing experiment, Relgraph performed best in most metrics. Relgraph-TransE and Relgraph-RotatE had better performances than baseline methods, such as TransE, RotatE, and ConvE. Relgraph-RotatE was 4.4% higher than the best algorithm before Relgraph, and Relgraph-TransE was 1.3% higher than the original TransE. The model version using relation graph (Relgraph-RotatE or Relgraph-TransE) also has advantages in performance compared to the version not used (GAT-RotatE or GAT-TransE).

Table 3. Results of drug repurposing experiment.

Algorithm	MRR	Hits@1	Hits@3	Hits@10
TransE	0.232	17.3	45.0	58.1
DistMult	0.164	16.0	30.3	34.8
ConvE	0.289	19.9	47.6	57.5
CompGCN	0.291	22.3	48.6	58.1
RotatE	0.296	22.8	49.1	57.9
GAT-TransE	0.234	17.5	45.1	58.6
GAT-RotatE	0.299	22.9	49.4	57.8
Ours:Relgraph-TransE	0.235	21.2	46.9	59.2
Ours:Relgraph-RotatE	0.309	23.5	50.3	59.5

4.3. Analytical Experiments

To gain a deeper understanding of the impact of various parameters in the model and delve into the inner workings of the model, we meticulously designed and executed a series of analytical experiments.

The impact of embedding dimension size. As Figure 4 demonstrates, we conducted an experimental analysis to investigate the impact of the embedding dimension of entities, predicates, and entity-links within Relgraph on its performance. The embedding dimensions of the model were set to 50, 100, 150, and 200, respectively. The results indicate that the performance improvement plateaus at a dimension of 200.



Figure 4. MRR of Relgraph under different embedding dimensions on FB15K-237 dataset.

The impact of training epoch. To analyze the learning efficiency of the Relgraph model, the MRR performance of each training epoch model was sampled during experiments on the FB15k-237 dataset, as displayed in Figure 5. It is evident that the model achieved satisfactory performance at approximately 500 epochs and reached optimal fitness at 3000 epochs.



Figure 5. The learning curve of Relgraph under different epochs on FB15K-237 dataset.

Comparison of schemes for calculating the rotation angles in the Relgraph-RotatE. To verify the effectiveness of the scheme to calculate rotation angles in the Relgraph-RotatE model as per Equation (23), we conducted a comparative experiment. We established two modified versions of the model, Relgraph-RotatE-min and Relgraph-RotatE-max, based on their distinct approaches to processing relation embeddings. The corresponding calculation formulas for rotation angles in RotatE are as follows:

$$p_r^{\theta,min} = \min(p_r^u, p_r^d) p_r^{\theta,max} = \max(p_r^u, p_r^d)$$
(23)

The performance comparison between the Relgraph-RotatE-min, Relgraph-RotatE-max, and Relgraph-RotatE models on the FB15k-237 dataset is displayed in Figure 6. It is evident that the rotation angle calculation scheme proposed in this paper offers several advantages.



Figure 6. Comparison of model performance under different rotation angle calculation schemes on the FB15k-237 dataset.

Comparison of different similarity algorithms in the Relgraph model. We also conducted an analysis and verification of the performance impact of different similarity algorithms in scoring functions (Equations (16) and (17)) on the Relgraph model. For this experiment, we utilized cosine similarity, F1 norm, and F2 norm in scoring functions to calculate the similarity between embeddings. The comparison of their performance on the FB15K-237 dataset is presented in Figure 7.



Figure 7. Comparison of model performance under different similarity algorithms on the FB15k-237 dataset.

5. Discussion

5.1. Performance Analysis of Relgraph

From the benchmark results presented in Table 1 and the drug repurposing experiment outcomes in Table 3, it is evident that the Relgraph outperforms the baseline model in terms of performance across various KGC tasks. This underscores the advantage of Relgraph in leveraging the relation graph for extracting information exchange between predicates.

Notably, the Relgraph exhibits the most significant performance improvement compared to baseline methods on the FB15K-237 dataset, as evident from Table 1. Improvements achieved by Relgraph-RotatE and Relgraph-TransE over suboptimal baseline methods amount to approximately 56.5% and 52.9%, respectively.

On the WN18RR dataset, the Relgraph exhibits a relatively smaller improvement over the baseline method. Among them, the Relgraph-RotatE approach exhibits the most notable improvement, reaching approximately 1.9%, followed by the Relgraph-TransE approach with approximately 0.4%.

Relgraph exhibits a significant performance improvement over the baseline method on the UMLS dataset. When compared to the suboptimal baseline, Relgraph-RotatE and Relgraph-TransE achieve improvements of 22.0% and 21.6%, respectively.

The Relgraph model exhibits an advantage compared to the baseline in its ability to identify new indications for existing drugs, achieving a performance improvement of 4.4% for Relgraph-RotatE relative to RotatE and 1.3% for Relgraph-TransE relative to TransE. This suggests that in large-scale knowledge bases, Relgraph remains effective.

Upon comparing different experimental tasks, it is observed that the Relgraph model exhibits its greatest advantage on FB15K-237, which has the highest number of predicates, and relatively least advantage on WN18RR with the lowest number of predicates. This underscores the unique advantage of the relation graph in handling complex KGs with numerous predicates.

5.2. The Performance Improvement Brought by the Relation Graph and the Universality of Relgraph

The results in Table 2 demonstrate that Relgraph outperforms ordinary GAT, regardless of the representation learning method used. Specifically, Relgraph-Rotate offers a 5.6% improvement over GAT-Rotate, while Relgraph-TransE offers a 5.9% improvement over GAT-TransE on the FB15K-237 dataset. This highlights the value of introducing relation graphs to capture predicate interactions.

The results in Table 2 also indicate that the Relgraph framework is versatile and can be seamlessly integrated with other transitive representation learning methods on KGs, enhancing their performance. Notably, the performance improvement of Relgraph-RotatE over the original RotatE is 56.5%, and the improvement of Relgraph-TransE over the original TransE is 72.3%. As Figure 3 shows, this improvement is consistent across datasets, highlighting the generalizability of the Relgraph framework for enhancing knowledge graph reasoning.

5.3. Analysis of Hyperparameters and Related Settings

In Figure 4, it can be observed that as the embedding dimensions in the Relgraph increase from 50 to 200, the model's performance improves accordingly. However, further increasing the dimensions does not significantly enhance the model's performance. Our experimental conclusion is that the optimal embedding dimension is dependent on factors such as the size of the dataset, the number of predicates, and the representation learning method used (TransE, RotatE, etc.).

As shown in Figure 5, the Relgraph typically achieves rapid convergence, approaching approximate optimal performance at around 1000 epochs, highlighting its relative efficiency. As the number of training epochs increases, the performance of the Relgraph continues to improve slightly.

The comparative experiment in Figure 6 demonstrates the effectiveness of the rotation angle calculation method (Equation (23)) utilized in RotatE. Compared to using max or min functions, Equation (23) exhibits a performance advantage of approximately 3%.

Based on the experimental results depicted in Figure 7, it can be concluded that in the Relgraph model, scoring triplets using cosine similarity yields the best performance. Conventionally, representation learning models for KGs utilize the F1 norm to measure vector similarity. The utilization of cosine similarity as the scoring function in Relgraph leads to optimal performance, possibly due to the influence of the relation graph. The predicate representation learned by the model is insensitive to vector magnitude but focuses more on vector angles.

6. Conclusions

This article proposes a new KG reasoning framework, Relgraph, which explicitly models the interaction between different relations by introducing a relation graph. An attention mechanism-based machine learning algorithm is designed to synchronously optimize GAT for original graph and relation GAT for relation graph, thereby improving the performance of transitive representation learning methods and multi-relational graph neural network models in reasoning tasks. The experimental results demonstrate the effectiveness of Relgraph in knowledge graph reasoning tasks. The universality of this framework and its ability to embed various representation learning algorithms make it widely applicable. We found that Relgraph is particularly suitable for reasoning on datasets with rich predicates, and can still perform KGC on large-scale datasets. The computational complexity of the model is on the same level as traditional GAT.

Limitations. The primary limitations of this article are twofold. Firstly, the graph attention learning mechanism upon which it relies is relatively conventional and lacks integration with the latest advancements in the field. Secondly, the proposed new framework lacks comprehensive exploration of its application potential, being confined primarily to the realm of transductive KGC, while neglecting other promising scenarios, such as inductive KGC, rule mining, and knowledge discovery.

Next, we will further explore the unique advantages of Relgraph in predicate information mining and rule mining. We will also attempt to integrate some new GAT mechanisms into our framework, striving to make new breakthroughs in interpretable and highly generalized reasoning.

Author Contributions: Conceptualization, X.T. and Y.M.; methodology, X.T.; software, X.T.; validation, X.T.; formal analysis, X.T.; investigation, X.T. and Y.M.; resources, X.T. and Y.M.; data curation, X.T. and Y.M.; writing—original draft preparation, X.T.; writing—review and editing, X.T. and Y.M.; visualization, X.T.; supervision, Y.M.; project administration, Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This study doesn't involve humans or animals.

Informed Consent Statement: Not applicable. This study doesn't involve humans.

Data Availability Statement: The FB15K-237 Knowledge Base Completion Dataset is available at https://www.microsoft.com/en-us/download/details.aspx?id=52312 (accessed on 15 February 2024). The UMLS Dataset can be downloaded at https://github.com/Colinasda/KGdatasets (accessed on 15 February 2024). The WN18RR Dataset can be downloaded at https://github.com/Colinasda/KGdatasets (accessed on 15 February 2024). The RTX-KG2 Dataset can be downloaded at https://github.com/Colinasda/KGdatasets (accessed on 15 February 2024). The RTX-KG2 Dataset can be downloaded at https://github.com/Colinasda/KGdatasets (accessed on 15 February 2024). The RTX-KG2 Dataset can be downloaded at https://github.com/RTXteam/RTX-KG2 (accessed on 15 February 2024).

Acknowledgments: The work and writing of this thesis have received strong support and assistance from Xin Wang from the Department of Computer Science at Tsinghua University.

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

References

- Li, H.; Wang, Y.; Zhang, S.; Song, Y.; Qu, H. KG4Vis: A knowledge graph-based approach for visualization recommendation. *IEEE Trans. Vis. Comput. Graph.* 2021, 28, 195–205. [CrossRef] [PubMed]
- Lin, Z.; Ke, H.; Wong, N.Y.; Bai, J.; Song, Y.; Zhao, H.; Ye, J. Multi-Relational Graph based Heterogeneous Multi-Task Learning in Community Question Answering. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 1–5 November 2021; pp. 1038–1047.
- 3. Zhu, Y.; Che, C.; Jin, B.; Zhang, N.; Su, C.; Wang, F. Knowledge-driven drug repurposing using a comprehensive drug knowledge graph. *Health Inform. J.* 2020, *26*, 2737–2750. [CrossRef] [PubMed]
- 4. Vashishth, S.; Sanyal, S.; Nitin, V.; Talukdar, P. Composition-based multi-relational graph convolutional networks. *arXiv* 2019, arXiv:1911.03082.
- 5. Peng, H.; Zhang, R.; Dou, Y.; Yang, R.; Zhang, J.; Philip, S.Y. Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks. *ACM Trans. Inf. Syst.* **2022**, *40*, 1–46. [CrossRef]
- 6. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* 2013, 26, 2787–2795.
- Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *arXiv* 2019, arXiv:1902.10197.
- 8. Balažević, I.; Allen, C.; Hospedales, T.M. Tucker: Tensor factorization for knowledge graph completion. *arXiv* 2019, arXiv:1901.09590.
- 9. Yang, B.; tau Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the ICLR (Poster), San Diego, CA, USA, 7–9 May 2015.
- 10. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2D Knowledge Graph Embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence , San Francisco, CA, USA, 4–9 February 2017.
- 11. Socher, R.; Chen, D.; Manning, C.D.; Ng, A.Y. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In Proceedings of the NIPS, Lake Tahoe, NV, USA, 5–10 December 2013.
- 12. Yue, L.; Zhang, Y.; Yao, Q.; Li, Y.; Wu, X.; Zhang, Z.; Lin, Z.; Zheng, Y. Relation-aware Ensemble Learning for Knowledge Graph Embedding. *arXiv* 2023, arXiv:2310.08917.
- Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* 2009, 20, 61–80. [CrossRef] [PubMed]
- Zhang, M.; Chen, Y. Link Prediction Based on Graph Neural Networks. In Proceedings of the NeurIPS, Montréal, QC, Canada, 3–8 December 2018.
- 15. Zhang, M.; Chen, Y. Inductive Matrix Completion Based on Graph Neural Networks. In Proceedings of the ICLR, Virtual, 26–30 April 2020.
- Nathani, D.; Chauhan, J.; Sharma, C.; Kaul, M. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
- Zhang, Y.; Zhou, Z.; Yao, Q.; Chu, X.; Han, B. Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 3446–3457.
- 18. Bi, Z.; Cheng, S.; Chen, J.; Liang, X.; Xiong, F.; Zhang, N. Relphormer: Relational Graph Transformer for Knowledge Graph Representations. *Neurocomputing* **2024**, *566*, 127044. [CrossRef]
- Toutanova, K.; Chen, D. Observed Versus Latent Features for Knowledge Base and Text Inference. In Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, Beijing, China, 26–31 July 2015.
- Bodenreider, O. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Res.* 2004, 32, D267–D270. [CrossRef] [PubMed]
- Wood, E.C.; Glen, A.K.; Kvarfordt, L.G.; Womack, F.; Acevedo, L.; Yoon, T.S.; Ma, C.; Flores, V.; Sinha, M.; Chodpathumwan, Y.; et al. RTX-KG2: A system for building a semantically standardized knowledge graph for translational biomedicine. *BMC Bioinform.* 2022, 23, 400. [CrossRef] [PubMed]
- 22. Xin, J.; Afrasiabi, C.; Lelong, S.; Adesara, J.; Tsueng, G.; Su, A.I.; Wu, C. Cross-linking BioThings APIs through JSON-LD to facilitate knowledge exploration. *BMC Bioinform.* **2018**, *19*, 30. [CrossRef] [PubMed]
- Kilicoglu, H.; Shin, D.; Fiszman, M.; Rosemblat, G.; Rindflesch, T.C. SemMedDB: A PubMed-scale repository of biomedical semantic predications. *Bioinformatics* 2012, 28, 3158–3160. [CrossRef] [PubMed]
- 24. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* 2014, arXiv:1412.6980.
- Meilicke, C.; Chekol, M.W.; Ruffinelli, D.; Stuckenschmidt, H. Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019; pp. 3137–3143.

- 26. Yang, F.; Yang, Z.; Cohen, W.W. Differentiable learning of logical rules for knowledge base reasoning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2316–2325.
- 27. Nguyen, D.Q.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. *arXiv* 2018, arXiv:1712.02121.

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