

## Article

# Link Prediction Based on Data Augmentation and Metric Learning Knowledge Graph Embedding

Lijuan Duan <sup>1,2,3,\*</sup> , Shengwen Han <sup>1,2,3</sup>, Wei Jiang <sup>4,\*</sup>, Meng He <sup>1,2,3</sup> and Yuanhua Qiao <sup>5</sup>

<sup>1</sup> Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; han\_sw@emails.bjut.edu.cn (S.H.); hem@emails.bjut.edu.cn (M.H.)

<sup>2</sup> Beijing Key Laboratory of Trusted Computing, Beijing 100124, China

<sup>3</sup> National Engineering Laboratory for Critical Technologies of Information Security Classified Protection, Beijing 100124, China

<sup>4</sup> Chinese Academy of Cyberspace Studies, Beijing 100048, China

<sup>5</sup> Faculty of Science, Beijing University of Technology, Beijing 100124, China; yuanhuaqiao@bjut.edu.cn

\* Correspondence: ljduan@bjut.edu.cn (L.D.); jw@bjut.edu.cn (W.J.)

**Abstract:** A knowledge graph is a repository that represents a vast amount of information in the form of triplets. In the training process of completing the knowledge graph, the knowledge graph only contains positive examples, which makes reliable link prediction difficult, especially in the setting of complex relations. At the same time, current techniques that rely on distance models encapsulate entities within Euclidean space, limiting their ability to depict nuanced relationships and failing to capture their semantic importance. This research offers a unique strategy based on Gibbs sampling and connection embedding to improve the model's competency in handling link prediction within complex relationships. Gibbs sampling is initially used to obtain high-quality negative samples. Following that, the triplet entities are mapped onto a hyperplane defined by the connection. This procedure produces complicated relationship embeddings loaded with semantic information. Through metric learning, this process produces complex relationship embeddings imbued with semantic meaning. Finally, the method's effectiveness is demonstrated on three link prediction benchmark datasets FB15k-237, WN11RR and FB15k.

**Keywords:** knowledge graph embedding; metric learning; link prediction; negative sampling; semantic extraction; relation fusion



**Citation:** Duan, L.; Han, S.; Jiang, W.; He, M.; Qiao, Y. Link Prediction Based on Data Augmentation and Metric Learning Knowledge Graph Embedding. *Appl. Sci.* **2024**, *14*, 3412. <https://doi.org/10.3390/app14083412>

Academic Editor: Douglas O'Shaughnessy

Received: 6 March 2024

Revised: 2 April 2024

Accepted: 3 April 2024

Published: 18 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

A knowledge graph is a vast storehouse of entities and their relationships, structured as triplets. Knowledge graph embedding (KGE) is a method for mapping entities and relations existing in a knowledge graph to the low-dimensional [1], dense space while preserving the ternary's internal structure and allowing for a dispersed representation of things and relations. Similar to tasks in [2–5], computing the low-dimensional vectors for entities and relations in a knowledge graph enhances recommendation accuracy, Q&A answers, and semantic representations. This improvement is crucial for enhancing the efficacy of knowledge graph applications and is a vital aspect of knowledge graph completion and link prediction. As a result, a growing multitude of scholars are concentrating on link prediction based on KGE.

Recently, researchers have introduced a range of KGE models to tackle link prediction tasks. Link prediction involves forecasting another entity that can complete a valid triple  $(h, r, t)$  when given one entity under a specific relation. Deep learning-based KGE models often establish a scoring function to decide the presence of a link based on the score of the projected triplet  $\Phi(h, r, t)$ . KGE models that focus only on internal facts include the Translation Distance Model (TDM) and Semantic Matching Model (SMM). The difference

between the two is that the former uses distance as the scoring function, and the latter uses the similarity measure as the evaluation function.

TDMs are widely used in link prediction tasks within the realm of KGE models due to their simplicity and efficiency. However, it is important to note that TDMs have limitations when it comes to expressiveness, as observed in [6]. Empirically, they tend to underperform when compared to other models, particularly in modeling intricate relation patterns. Compared with other methods, SMM has excellent performance in both result performance and theoretical guarantee. Nonetheless, the substantial number of parameters in SMMs presents obstacles when it comes to deploying them in extensive knowledge graphs within industrial contexts. In summary, the challenge lies in achieving robust modeling capabilities for intricate relation patterns simultaneously while maintaining performance in KGE learning.

Meanwhile, because the knowledge graph only contains positive samples, in order to assure the model's generalization capacity, negative examples must be generated throughout the training process using a tailored technique as a data augmentation strategy to increase the model's training effect. Currently, the widely adopted method of negative sampling randomly selects negative samples by substituting either the head or tail entity in the triplet. However, as training progresses, an abundance of zero-loss negative samples emerges, leading to gradient disappearance due to the challenge of adjusting these negatives effectively for each positive sample. Furthermore, negative sampling should be considered to avoid generating erroneous negative triples. For example, for *(Beijing, located in, China)*, generating pseudo-negative examples by randomly replacing the head entity with *(Shanghai, located in, China)* will not only affect the model training effect, but may also cause test set leakage. As a result, generating high-quality negative samples is also a pressing issue.

To tackle the limitations of existing TDMs in obtaining semantic information and the challenges posed by SMMs with an excess of parameters, we introduce a novel KGE model. This approach dynamically represents entities and comprises two modules: Relation Fusion and Metric Learning (ML). The ML module optimizes learning efficiency by extracting shared parameters from the complete KG, inspired by prior work [7,8]. This minimizes model parameters while enhancing semantic information content. Additionally, we propose a technique involving Relation Cutting and the development of a Mapping Matrix, influenced by [9,10].

The core aim of this strategy is to amplify the differentiation between head and tail entities while maintaining the coherence of identical triplets. This is accomplished by dynamically embedding head and tail entities onto their respective relational hyperplanes, employing the associated projection matrices. Additionally, the relation fusion module bolsters the effectiveness of relational embeddings by amalgamating pertinent vector information associated with relations. As a result, this diminishes the semantic gap within entities and their associated negative samples.

In the context of the research, we contribute the following to address this challenge:

- We introduce the concept of Gibbs sampling and present a path-based negative sampling method to generate high-quality negative samples. This method enhances the generalization capacity of relation embeddings.
- To overcome the limitations of prior models in handling intricate relations, we introduce a technique to acquire unique embeddings for a given entity across diverse relations and positions. This dynamic representation effectively captures complex relationships.
- To lower the quantity of parameters simultaneously while maintaining overall features, we introduce the concept of shared feature tensors. This improvement enables the model to perform effectively across several embedding dimensions and considerably improves training efficiency.

This article represents an extended version of a paper [11] presented at the ICCAI conference. In this iteration, we incorporate a Gibbs negative sampling module and auxiliary loss. We expand the experimental validation and results analysis, demonstrating the efficacy of the newly introduced modules.

## 2. Related Work

### 2.1. Data Augmentation for Link Prediction Task

Data augmentation is a useful method for expanding dataset size and enhancing data diversity. Recently, techniques employing generative adversarial networks to generate textual descriptions in knowledge graphs have emerged. OnToZSL [12] developed an ontology encoder to learn relationship representations by encoding intricate logical information between textual descriptions in knowledge graphs. However, this approach solely focuses on generating textual descriptions, overlooking the potential benefits of incorporating an entity's structural information to enrich knowledge graph data samples. WRAN [13] leveraged adversarial networks to learn entity embeddings with invariant relationships from an entity's textual features and structural encodings. With the advancement of natural language corpora, various completion methods have utilized pre-trained models to tackle issues. KG-BERT [14], for instance, is a BERT-based knowledge graph representation learning method that achieves more accurate and expressive entity and relationship representations by utilizing context-aware word vectors. Despite these advancements, the challenge of insufficient knowledge graph samples remains inadequately addressed.

Negative sampling was put forward by [15]. It was originally used to speed up the training of the Skip Gram model. Now, it has been widely used in CV (Computer Vision), recommendation systems, NLP (Natural Language processing), and other fields. In the field of knowledge graph, Ref. [16] demonstrated that negative sampling is equally crucial with positive sampling and experimentally negated the conventional intuition in graph representation learning that "sampling adjacent nodes as positive node pairs and sampling distant nodes as negative node pairs".

In the knowledge map embedding task, negative sampling is regarded as an important module that affects model performance. Take the TransE model [9] as an illustration; its objective function is defined as the gap between positive and negative samples. GANs (Generative Adversarial Networks) have also been introduced for generating negative samples. After a detailed analysis of static negative sampling methods, KBGAN [17] found that the majority of negative samples are weak negative examples rather than strong negative examples. During training, the scoring of these negative examples rapidly decreases, leading to the problem of gradient vanishing. By incorporating GANs [18], one can generate well-crafted negative sample triplets and leverage new distributions in the training process. IGAN [19] designed a generator and performed advanced negative sampling based on softmax, using the boundary loss training of positive and negative examples in the discriminator to obtain the final embedded model. A knowledge-guided attention mechanism was introduced by [20], and they devised a semantic sampler along with comparative learning to acquire diverse representations for negative samples across various relationships. NSCaching [21] designed two modules: head entity cache and tail entity cache. During the procession of training, to replace true triplets, entities are selected from the head entity cache and tail entity cache. In addition, the two cache modules are also updated as the model is trained.

In addition, there are some semantic related negative sampling models, such as joint relationship context negative sampling, entity similarity-based negative sampling, etc. As the performance of models encounters limitations in link prediction, researchers are exploring solutions to improve embedding quality from multiple perspectives beyond model design. Starting from the underlying logic of embedded models, negative sampling is undoubtedly a research perspective with great potential.

## 2.2. Translational Distance Models

The TransE model proposed by [22] represents a seminal work in translational distance models. It operates on the assumption that if a triple  $(h, r, t)$  exists, the vector representations of the head and tail entities, denoted as  $h$  and  $t$ , along with the relation  $r$ , should approximately fulfill the constraint  $h + r \approx t$ . This simple constraint yields a straightforward and effective modeling approach. However, it can sometimes result in the convergence of representations for semantically different entities in vector space. After being trained by TransE, entities like (Shanghai, located in China) and (Beijing, located in China) may exhibit vector representations  $\mathbf{v}_{\text{Shanghai}} \approx \mathbf{v}_{\text{Beijing}}$ , even though these entities should not possess such a high similarity.

To overcome challenges, TransH, introduced by [9], utilizes hyperplanes. Independently projecting the head and tail entities onto the relation-associated hyperplane precedes the transformation process. In contrast, RotatE, proposed by [23], utilizes rotation operations in a complicated space, departing from operations of summing in Euclidean space. This enhances the model's capability to handle symmetry relations. PairRE, proposed by [10], leverages pairwise relationships to model complex relations effectively.

Traditional TDMs, including TransE, come with their merits, such as straightforward operations, a reasonable parameter count, and effective learning. However, they often fall short in capturing the full semantic richness of learned vector representations. In our approach, we seek to augment the modeling capacity beyond what TransH offers by introducing paired additional mapping matrices. This augmentation allows for a more sophisticated representation of intricate relations while preserving the simplicity and efficiency found in TDMs.

## 2.3. Semantic Matching Models

In the realm of knowledge graph applications, semantic matching models hold indispensable significance. An eminent contender in this class is RESCAL proposed by Nickel et al. [24], which characterizes a knowledge graph with a binary three-dimensional tensor  $\mathcal{X} \in \mathbb{R}^{|E| \times |E| \times |R|}$  and depends on one-hot encoding. While RESCAL offers a foundational approach, it has limitations. It fails to capture the rich semantic nuances within the graph and scales poorly with the expansion of the size of the knowledge graph, making it less suitable in industrial-scale scenarios. Several variations of RESCAL have emerged to address these shortcomings. DistMult [25] simplifies the RESCAL model by constraining the relation matrix to a diagonal form, effectively reducing the overall parameter complexity. Going a step further, ComplEx [26] incorporates complex embeddings, enabling a more accurate representation of non-symmetric relations. TuckER [27] employed a decomposition algorithm to break down the tensor into entity and relation matrices, offering a holistic approach to semantic matching. ConvE [28] takes a unique perspective by separately reconstructing head entity and relation embeddings and applying a 2D convolution layer to extract semantic features, enhancing its capability to handle complex relations. InteractE [29] identifies a limitation in previous models, particularly ConvE, which restricts feature interaction between entities and relations. To improve completion performance, InteractE prioritizes increasing feature interaction, yielding better results.

While these semantic matching models have significantly advanced knowledge graph applications, they often come with a high parameter count, posing challenges in terms of overfitting and scalability. Inspired by TuckER, we propose the introduction of a shared parameter tensor to mitigate these issues, allowing for a more effective learning in large-scale knowledge graph scenarios.

In summary, the evolution of semantic matching models has significantly enriched the landscape of knowledge graph-based applications. Each model offers unique insights and improvements over its predecessors, and our proposal of shared parameters seeks to further enhance efficiency and scalability in this domain.

#### 2.4. Metric Learning

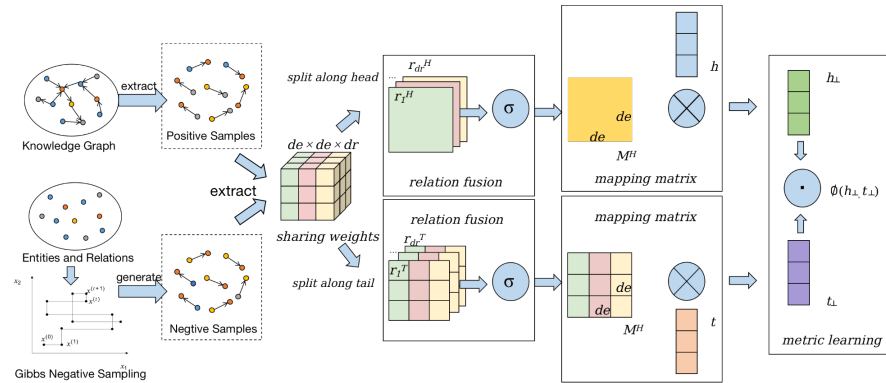
Metric learning is a method of spatial mapping that enables the acquisition of a feature space. Within this space, all data are transformed into feature vectors, where the distance between feature vectors of similar samples is minimized while that between dissimilar samples is maximized, thereby facilitating data discrimination.

Metric learning can be categorized into two types based on whether the original feature space is transformed before measurement: direct measurement and transformed measurement. Direct measurement typically relies on the Euclidean distance between objects in the original feature space, such as the K-Nearest Neighbor (KNN) algorithm [30], which calculates the similarity between objects directly based on Euclidean distance. Transformed measurement, on the other hand, is generally based on the Mahalanobis Distance, which measures distances after projecting source vectors onto a projection matrix. The metric function parameterizing the KNN classifier introduces the notion of a large-margin nearest neighbor classifier. The Mahalanobis Distance is an extension of the Euclidean distance, encompassing all possible metrics in the feature space under linear conditions. Consequently, learning the metric function translates into learning an embedding representation by transforming input features into a lower-dimensional embedding space, thus circumventing the degradation issues of conventional metric functions such as Euclidean distance in high dimensions.

Ref. [31] introduced the Contrastive Loss function, which trains the model to determine whether each pair of faces comes from the same person, aiming to pull similar samples closer while pushing dissimilar samples apart. Ref. [32] proposed the Triplet Loss function as an improvement, where each input consists of an anchor sample, a positive sample, and a negative sample, and the network judges the distance between the anchor sample and the positive and negative samples. Due to the similarity between small-sample classification problems and face recognition tasks, these methods have gradually been applied to some classification problems. For instance, Ref. [33] proposed Siamese Networks, which utilize the Contrastive Loss for single-sample image classification. Ref. [34] implemented self-supervised image classification based on the Siamese Network architecture by treating any image and its transformation as a positive pair and any randomly selected pair of images as negative samples.

### 3. Methodology

In this chapter, we present our proposed methodology, starting with the introduction of a negative sampling technique aimed at obtaining superior negative samples. Secondly, a knowledge graph embedding method is proposed that combines relationship fusion and metric learning [11]. This model, which has fewer parameters and integrates semantic information, is used in conjunction with the two methods introduced for link prediction tasks. First, in Section 3.1, we introduce the proposed negative sampling method. Next, Sections 3.2 and 3.3 present the proposed metric learning module and relation fusion module, respectively. Finally, in Section 3.4, we delve into the objective function, while the model's comprehensive architecture is depicted in Figure 1.



**Figure 1.** Overall architecture of the model. Initially, entities are chosen from the knowledge graph for Gibbs sampling to derive high-quality negative sample triplets. Subsequently, shared weights between positive and negative samples are extracted, and the samples are partitioned separately along the directions of head and tail entities, resulting in two directional mapping matrices. Finally, the similarity between these mapping matrices is measured through metric learning.

### 3.1. Negative Sampling

In this section, we introduce the negative sample generation module based on Gibbs negative sampling. Moreover, in the second subsection, we introduce our relation negative sampling method.

#### 3.1.1. Gibbs Negative Sampling

Gibbs negative sampling is introduced to tackle the challenge of a substantial rise in zero-loss negative samples caused by the subpar quality of negatives, resulting in gradient vanishing. It also addresses the issue of test set leakage. Ref. [16] utilizes the Metropolis–Hastings algorithm for negative sample generation in graph representation learning, which is well-suited for one-dimensional distributions. However, in the context of embedding knowledge maps, entities always appear in pairs within triples, manifesting as head entities and tail entities. Consequently, a two-dimensional distribution is more appropriate for generating samples. Drawing inspiration from this, our study explores Gibbs sampling, commonly used for multivariate joint distributions, as it offers a simpler approach. Specifically, we investigate the distribution characteristics of entities in the knowledge map. For the entity–relation pair  $(h, r)$ , the negative sample entities of higher quality are those  $T'$  entities that exist in the observed facts in the form of  $(\sim, r, t')$ . The formula for this is given in Algorithm 1.

---

#### Algorithm 1 Gibbs Negative Sampling Algorithm

---

**Require:** Conditional probability distribution  $P$ , state transition times threshold  $n_1$ , number of negative samples  $n_2$

**Require:** Random initialization state values  $x_1^{(0)}, x_2^{(0)}$

**for**  $\mu$  in Search Grid **do**

    Sample  $x_2^{(t+1)}$  from the conditional probability distribution  $P(x_2 | x_1^{(t)})$

**for**  $t = 1, 2, \dots, n_1 + n_2 - 1$  **do**

        Sample  $x_1^{(t+1)}$  from the conditional probability distribution  $P(x_1 | x_2^{(t+1)})$

**end for**

    Calculate loss  $L_\mu$  of sample set:

$$\{(x_1^{(n_1)}, x_2^{(n_1)}), (x_1^{(n_1+1)}, x_2^{(n_1+1)}), \dots, (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)})\}$$

**if** current loss is greater than historical maximum loss **then**

        Replace the maximum loss with the candidate sample set

**end if**

**end for**

**Output:** Sample set

$$N_{(h,r)} = \{(x_1^{(n_1)}, x_2^{(n_1)}), (x_1^{(n_1+1)}, x_2^{(n_1+1)}), \dots, (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)})\}$$


---

### 3.1.2. Relation Negative Sampling

This section presents a path-based approach to negative sampling, intending to enhance the expressive capabilities of relations. The method operates as follows for a given relation: initially, all head and tail entities are connected to the relation to create a set of entities:

$$\epsilon_{r_i} = \{h \mid (h, r_i, t) \in T\} \cup \{r \mid (h, r_i, t) \in T\} \tag{1}$$

Pick the relation linked to every entity within the entity set  $\epsilon_{r_i}$  to create a negative relation sample:

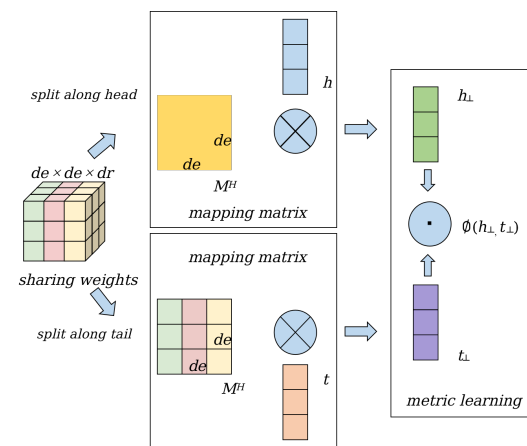
$$N_{r_i} = \{r \mid (h, r, t) \in T \wedge h, r \in \epsilon_{r_i}\} \tag{2}$$

Within this context,  $T$  represents the set of triplets in the knowledge graph, and  $N_{r_i}$  denotes the set of negative relation samples, representing the relations designated for comparison.

### 3.2. Metric Learning Module

This section presents an embedding methodology grounded in entity metric learning. Metric learning focuses on the distance relationship between data entities, measuring the similarity between them [7,8]. In contrast to TDMs, which utilize Euclidean distance for entity translation, we propose that enhancing embeddings through the measurement of semantic similarity between entities can yield more expressive embeddings.

In this process, the matrix functions to project the head and tail entities onto a dedicated hyperplane determined by the relation. In this process, the matrix projects entities onto a dedicated hyperplane determined by the relation. Following this projection, we derive the final entity and relation representations by minimizing the cosine similarity between these representations. The process is illustrated in Figure 2.



**Figure 2.** The overall architecture of the metric learning module.

#### 3.2.1. Shared Parameters

In a specific graph scenario, the entity features and relation features are constrained. As a consequence of this, a KG can be represented as two dimensions:  $de$  and  $dr$ .  $de$  represents the dimension of all entities in the graph, while  $dr$  represents the dimension of all relations. Moreover, the entities, relations, and the head and tail inhabit distinct feature spaces. The KG features are critical to embedding quality. Utilizing feature information poses a challenge because the available data often lack a comprehensive representation of these features.

Referencing the literature [23,35], this paper expresses relation features in a vector space of dimensionality  $dr$  and entity features in a vector space of dimensionality  $de$ .

We adopt an implicit feature representation approach, utilizing a three-dimensional tensor  $W \in \mathbb{R}^{de \times de \times dr}$  to capture feature interactions. Compared with previous methods, our approach avoids explicitly expressing the eigenvalues and embedding dimensions of entities and relationships, mitigating the exponential growth of parameters while the

various relations and entities scale. Considering the potential disparity in vector spaces between head and tail entities, directly measuring their distance lacks practical relevance. As a result, a projection matrix must be obtained in order to map them in a specified space before performing measurements. The projection matrix in this work is relation-based-embedded.

The entity’s final embedding is dynamically determined by relational embedding to achieve unique representations of the same entity across various relations and locations.

Unlike previous works, we propose a new method: the relation  $r$  in this paper is split into two components:  $[r^H, r^T]$ , with  $r = r^H || r^T$ , where  $||$  denotes vector concatenation. By multiplying each pair of relation vectors with the shared parameter tensor, we derive the projection matrix for both the head and tail entities, illustrated in Figure 3.

$$M^H, M^T = r^H \times W, r^T \times W^T \tag{3}$$

where  $M^H, M^T \in \mathbb{R}^{de \times de}$  denote the projection matrices for the head and tail entities, and  $W, W^T \in \mathbb{R}^{de \times de \times dr}$  represents the shared parameter tensor, with  $W^T$  being the transposed form of  $W$ . The goal is to maintain relational consistency by providing different representations for a given entity at the head and tail positions. After obtaining the projection matrix, we compute the final entity representation in the context of the current relation.

$$h_{\perp}, t_{\perp} = hM^H, tM^T \tag{4}$$

The method here does not employ distance metrics such as Euclidean distance because the head and tail entities may not lie on the same plane. Directly using distance metrics would not make sense in such cases. Therefore, the method of distance measurement after projection was chosen.

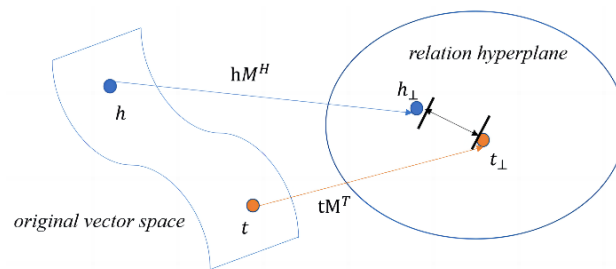


Figure 3. Vector projection diagram.

### 3.2.2. Metric Learning

Metric learning is a method of spatial mapping that enables the learning of a feature space. In this space, all data are transformed into feature vectors, with similar samples having feature vectors close together and dissimilar samples having feature vectors far apart, resulting in a distinct separation between them. Considering that entities should manifest markedly different feature information under distinct relationships and positions, this paper projects the head and tail entities of the triplets onto hyperplanes determined by relationships. Through measuring the semantic similarity of projected head and tail entities, distinctly separate embedding representations are learned for entities under different relationships. This allows for clearer distinctions between entities, leading to enhanced model accuracy.

The semantic affinity amid the projected entities is evaluated to assess whether they correspond to a specific relation, thus determining the validity of the resulting triple. This evaluation applies the cosine similarity between the head and tail entities, resulting in a triplet score, defined as follows:

$$\phi(h, t) = \frac{\sum_{i=1}^n h_i \times t_i}{\sqrt{\sum_{i=1}^{de} (h_i)^2} \times \sqrt{\sum_{i=1}^n (t_i)^2}} \tag{5}$$



Here,  $de$  represents the dimension of the entity vector. We apply a *sigmoid* to calculate the similarity score to obtain the predicted value.

$$p = \frac{1}{1 + e^{-\phi(h_{\perp}, t_{\perp})}} \tag{6}$$

Finally, the model utilizes binary cross-entropy as the loss function for optimization, calculated as follows:

$$L(p, y) = -\frac{1}{n_e} \sum_{i=1}^{n_e} (y^i \log(p^i) + (1 - y^i) \log(1 - p^i)) \tag{7}$$

In the given formula,  $y^i$  represents whether there is a triplet  $i$  in reality,  $y^i \in \{0, 1\}$ . Additionally,  $n_e$  denotes the number of entities in the KG.

### 3.3. Relation Fusion Module

In this section, we present a network layer that leverages an attention mechanism for refining relation representations. This improvement is achieved through the aggregation of vector representations from other relations. Prior research has predominantly focused on entity embedding learning, often considering relation embedding learning as relatively straightforward. Common operations like weight matrices have been employed to update relation embeddings. Therefore, our study delves into methods to enrich the expressiveness of relation embeddings and suggests that relations should incorporate pertinent information from other relations to enhance their semantics. The schematic diagram is shown in Figure 4.

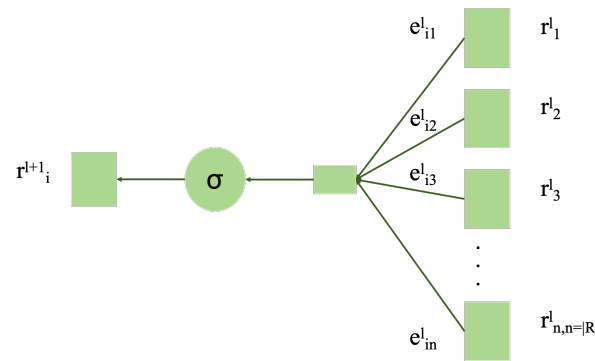


Figure 4. Relation fusion.

Ref. [36] conducted an analysis of the capabilities of various existing models in handling the aforementioned relations. They pointed out that these methods were unable to simultaneously model the upper two relations' patterns. To address this limitation, our study explores the potential of simultaneously modeling two relations by merging them together. Drawing inspiration from the graph attention network, which has demonstrated expressive capabilities [37–39], we employ a fusion operation in this paper.

If the relations  $r_1$  and  $r_2$  satisfy  $r_2(e_1, e_2) \Rightarrow r_1(e_2, e_1)$ , then these relationships are considered a pair of inverse relations, where  $e_1$  and  $e_2$  represent two entities. If the relations  $r_1, r_2,$  and  $r_3$  satisfy  $r_2(e_1, e_2) \wedge r_3(e_2, e_3) \Rightarrow r_1(e_1, e_3)$ , then these relationships are a composed relation.

In order to obtain the attention coefficient between two relations as  $j$  and  $i$ , we quote Formula (5):

$$e_{ij} = \text{LeakyReLU}[\phi(r_i, r_j)] \tag{8}$$

After establishing the semantic meaning shared between relations, the aggregation of related relation embeddings is performed based on the attention coefficient, as follows:

$$r_i = \sigma\left(\sum_{r_j \in R_i} e_{ij} r_j\right) \quad (9)$$

The aggregated embeddings in this step are used to update the previous relational embeddings.  $\sigma(\cdot)$  represents *sigmoid* and  $i$  represents the sequence number of a relationship selected for fusion, which contains the two special relation patterns mentioned above.

### 3.4. Loss Function

We primarily discuss the loss function of the proposed model in this section. Additionally, to enhance the model's performance, we introduce a triplet auxiliary loss. Based on Formula (5), the loss of a negative sample of relations in Section 3.1.1 is calculated by

$$L_{r_i} = -\frac{-1}{|\epsilon_{r_i} \cup N_{r_i}|} \log \frac{\sum_{r_i^+ \in \epsilon_{r_i}} \exp(\phi(r_i, r_i^+) / \tau)}{\sum_{r_i^- \in N_{r_i}} \exp(\phi(r_i, r_i^-) / \tau)} \quad (10)$$

In the above equation,  $\phi(\cdot)$  is defined in Formula (5), and  $\tau$  is a hyperparameter. Based on (7) and (10), compute the overall loss of the model:

$$L = L(p, y) + \sum_{i=1}^{n_r} L_{neg(i)} \quad (11)$$

where  $n_r$  denotes the overall quantities of relations in the KG.  $\sum_{i=1}^{n_r} L_{neg(i)}$  represents the loss of the relation fusion module, and  $L(p, y)$  signifies the metric learning module loss.

To further enhance the positive impact of the negative sample generation method on the model, this paper introduces an auxiliary loss module for the generator, employing the contrast loss function as the primary auxiliary loss. The contrast loss stands as a prevalent learning metric, which is frequently employed to construct a feature space that brings akin samples into proximity and simultaneously separates distinct samples. By incorporating this loss function, our objective is to increase the distance between the actual tail entity and other entities in the semantic space for the missing triple  $(h, r, ?)$ . This, in turn, aims to improve the accuracy of the embedding vector in the link prediction task. The corresponding loss function is as follows:

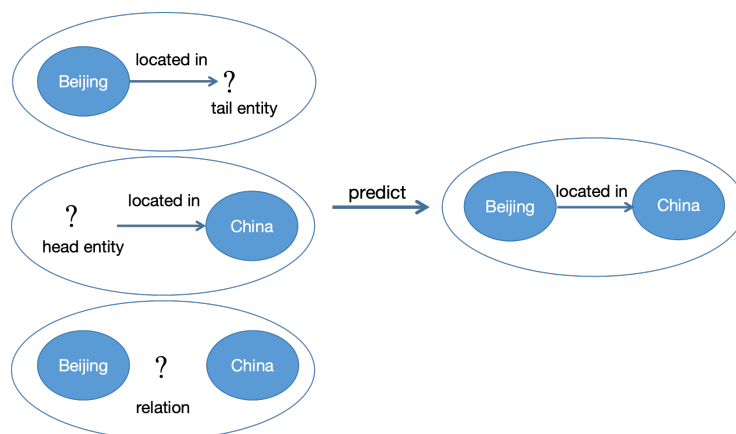
$$L(e_i, e_i^-) = -\frac{-1}{|Pos(i)|} \log \frac{\sum_{e_i^- \in N_{(h,r)}} \exp(\phi(e_i, e_i^-) / \tau)}{\sum_{e_j \in Pos(i)} \exp(\phi(e_i, e_j) / \tau)} \quad (12)$$

where  $Pos(i)$  represents all entities in the true triplets of a training batch,  $\phi(\cdot)$ . Use the cosine similarity function introduced by Formula (3).

## 4. Experiments

In this section, we describe experiments that were conducted on three established link prediction benchmarks in order to evaluate the effectiveness of our model from different perspectives. We refer to the model consisting of the Metric Learning and Relationship Fusion modules as KGE-EML [11].

In knowledge graphs, there may be instances of missing entities in triplets, such as for a triplet  $(h, r, ?)$ , where the tail entity is missing. Our experiments predict the missing tail entity using the knowledge graph embedding method proposed earlier, a task known as link prediction. For example, for the missing triplet  $(Beijing, located in, ?)$ , we predict the missing tail entity *China* using the embedding model proposed earlier in the text. Certainly, the missing triplet could also involve the absence of a relation or head entity, as illustrated in Figure 5.



**Figure 5.** A sample for a link prediction task in a knowledge graph. The ? indicates the missing entity or relation in the tuple.

#### 4.1. Datasets

We conducted experiments on three datasets: FB15k, FB12k-237, and WN11RR. FB12k-237 and WN11RR were created by removing inverse relations and duplicate relations from FB15k and WN11, respectively. By benchmarking the performances of the proposed model on each of these datasets against other models, comparable datasets, we critically evaluated its generalization ability.

In FB15k-237, there is an additional restriction: if a triplet  $(h, r_i, t)$  exists in the training set, the same triplet will not appear in the test or validation sets under  $(h, r_i, t)$ . WN18RR, on the other hand, does not have this restriction. Furthermore, we conducted experiments on the FB15k dataset to assess our model's ability to handle complex relations. Detailed information about these three datasets is provided in Table 1.

**Table 1.** Statistics of public datasets: FB15k, FB12k-237, and WN11RR.

| Dataset   | Entity | Relation | Train   | Valid  | Test   |
|-----------|--------|----------|---------|--------|--------|
| FB15k     | 14,951 | 1345     | 483,142 | 50,000 | 59,071 |
| WN11RR    | 40,493 | 11       | 86,835  | 5000   | 3134   |
| FB15k-237 | 14,541 | 237      | 272,115 | 17,535 | 20,466 |

#### 4.2. Link Prediction Results

In comparison to TDMs and SMMs, our model's link prediction results on FB15k-237 are presented in Table 2. Our model outperforms other TDMs and SMMs, indicating its promising performance. Notably, semantic matching models tend to perform better than distance models, as most of them utilize cross-entropy loss. In this paper, we also adopt such loss, treating all dataset entities as candidates for corrupted triplets.

Table 2 also showcases the model's performance on the WN18RR dataset. Since this dataset comprises only 11 relations, and our model primarily focuses on enhancing complex relation modeling, the advantage is less pronounced. However, our model excels in MRR and Hit@1 compared to all comparison models.

It is worth mentioning that our model's Mean Rank (MR) performance on the FB15k-237 dataset significantly surpasses that on the WN11RR dataset. This discrepancy can be attributed to the fact that the FB15k-237 dataset has a smaller total number of entities compared to WN11RR (specific data in Table 1). The increased number of entities in WN11RR introduces noisy entities, which can potentially impact the accuracy of the prediction results.

**Table 2.** Link prediction results on FB15k-237 and WN18RR.

| Model               | FB15k-237    |              |              |             | WN18RR       |              |              |              |
|---------------------|--------------|--------------|--------------|-------------|--------------|--------------|--------------|--------------|
|                     | MRR          | H@1          | H@3          | H@10        | MRR          | H@1          | H@3          | H@10         |
| TransE              | 0.33         | 0.231        | 0.369        | 0.528       | 0.223        | 0.014        | 0.401        | 0.529        |
| RotatE              | 0.338        | 0.241        | 0.375        | 0.533       | 0.476        | 0.428        | 0.492        | <b>0.571</b> |
| PairRE              | 0.351        | 0.256        | 0.387        | 0.544       | -            | -            | -            | -            |
| DistMult            | 0.308        | 0.219        | 0.336        | 0.485       | 0.439        | 0.394        | 0.452        | 0.533        |
| ComplEx             | 0.323        | 0.229        | 0.353        | 0.513       | 0.468        | 0.427        | 0.485        | 0.554        |
| TuckER              | 0.358        | 0.266        | 0.394        | 0.544       | 0.47         | 0.43         | 0.482        | 0.526        |
| ConvE               | 0.325        | 0.237        | 0.356        | 0.501       | 0.43         | 0.4          | 0.44         | 0.52         |
| InteractE           | 0.354        | 0.263        | -            | 0.535       | 0.463        | 0.43         | -            | 0.528        |
| PROCRUSTES          | 0.345        | 0.249        | 0.379        | 0.541       | 0.474        | 0.421        | 0.502        | 0.569        |
| <b>KGE-EML [11]</b> | 0.36         | <b>0.267</b> | 0.395        | 0.545       | 0.477        | 0.444        | 0.492        | 0.534        |
| <b>KGE-EML+GNS</b>  | <b>0.365</b> | 0.265        | <b>0.398</b> | <b>0.55</b> | <b>0.487</b> | <b>0.447</b> | <b>0.506</b> | 0.569        |

The bold text here in the “Model” column represents our proposed model, while under each evaluation metric, it signifies the best results.

#### 4.3. Capacity for Handling Complex Relations

In this study, we conducted experiments using the FB15k dataset, which contains a diverse range of relations, to assess our model’s capability in handling complex relations. Complex relations are defined in reference [6].

The results in Table 3 clearly demonstrate that our method outperforms other methods while dealing with relations in 1-to-n and n-to-1. This superiority arises from our method’s capability to address the limitations of distance-based models when handling such relations. We achieved this by partitioning the relation to obtain representations for positions of entities, facilitating a more effective modeling of these intricate relations.

**Table 3.** Capacity for handling complex relations: we conducted the experiment on FB15k and show the Hit@10 results in the table.

| Model       | 1-to-1       | 1-to-n       | n-to-1       | n-to-n       |
|-------------|--------------|--------------|--------------|--------------|
| ComplEx     | <b>0.939</b> | 0.896        | 0.822        | 0.902        |
| TransE      | 0.887        | 0.822        | 0.766        | 0.895        |
| RotatE      | 0.923        | 0.84         | 0.782        | 0.908        |
| PaiRE       | 0.785        | 0.899        | 0.872        | 0.94         |
| <b>Ours</b> | 0.865        | <b>0.948</b> | <b>0.875</b> | <b>0.913</b> |

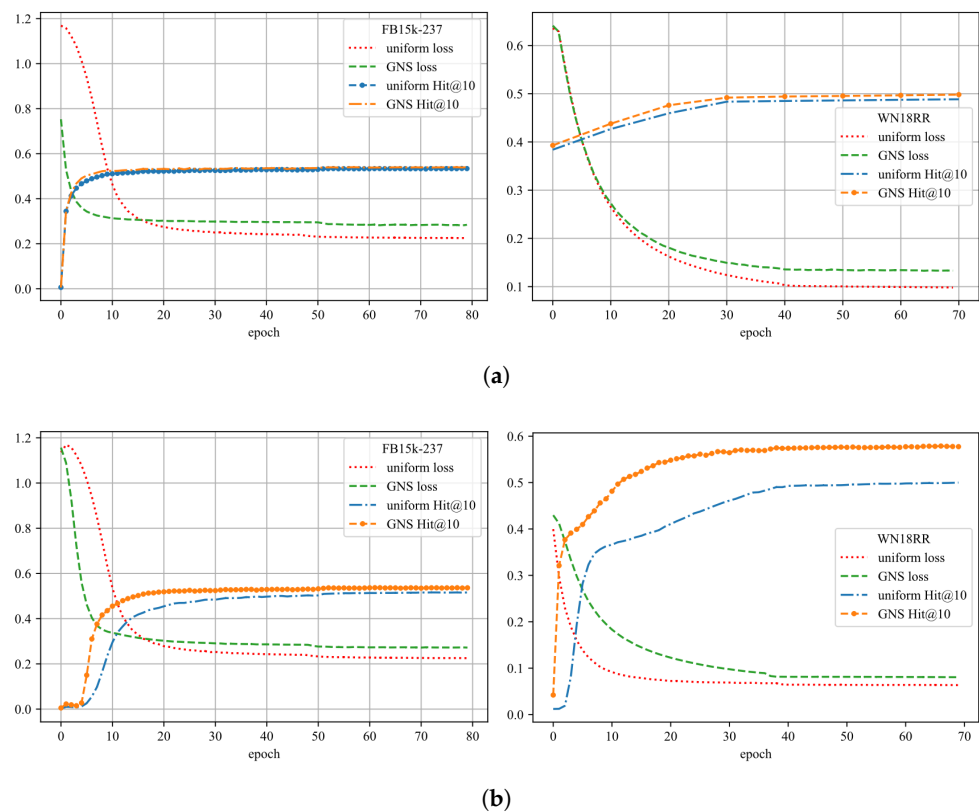
The bold text here in the “Model” column represents our proposed model, while under each complex relation, it signifies the best results.

#### 4.4. Comparison with Other Negative Sampling Methods

Due to the lack of special requirements for the scoring function in the model discriminator, we used scoring functions from multiple models as discriminators. Four negative sampling methods, namely, uniform random negative sampling, KBGAN, NSCaching, and EANS [40], were used to make comparisons with the negative sampling method proposed in this paper to verify the effectiveness of this method. The experimental results are shown in Table 4.

In Table 4, Uniform, KBGAN, NSCaching, and EANS [40] represent the current state-of-the-art negative sampling methods. This study utilized these methods along with the GNS method proposed herein to examine the impact of different negative sampling techniques on the model’s link prediction performance across distance models and tensor decomposition models. The results of the other methods in the table were sourced from published articles on the respective models, with bold fonts indicating the highest outcome. Upon scrutiny of the results provided in the table, it becomes evident that GNS consistently outperforms alternative negative sampling methods across the majority of models.

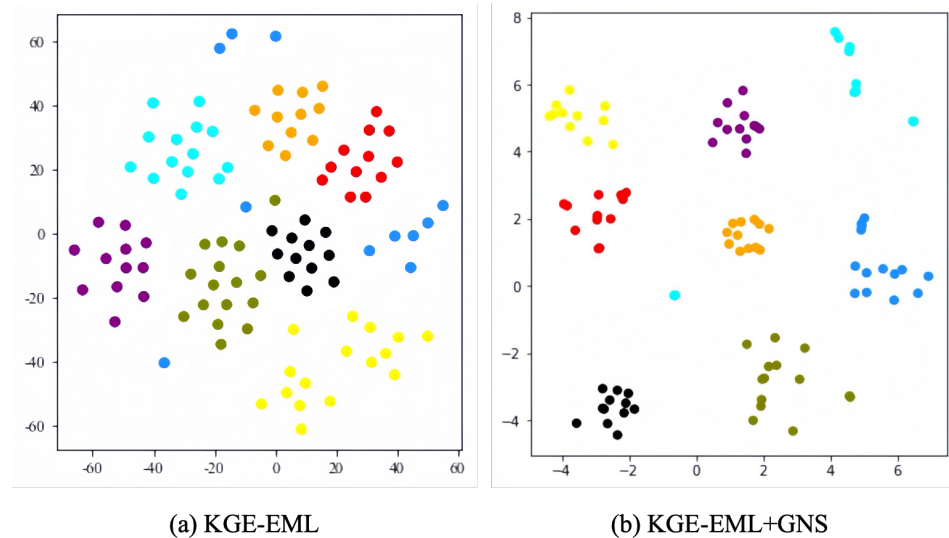
In order to verify whether the gradient vanishing problem is mitigated, this study used the uniform sampling method UNIFORM and the GNS method proposed in this paper to train the model, respectively; then, we plotted the change curves of the training loss and the accuracy index Hit@10 as the training proceeded. The experiments were conducted on TransE and the proposed model using two datasets, FB15k-237 and WN18RR, and the results are shown in Figure 6. First, it can be seen that although the losses of GNS and Uniform converge gradually, they do not reduce to zero because the small sampling process introduces noise into the gradient. However, due to the use of a negative sampling scheme based on Gibbs sampling, which is able to continuously generate high-quality negative samples as the model is trained, the loss of GNS ultimately remains greater than that of Uniform, and the model trained using GNS consistently outperforms Uniform on Hit@10.



**Figure 6.** Changes in loss values at different stages of GNS and Uniform training. (a) TransE; (b) KGE-EML [11].

Figure 7 displays the results of embedding visualization before and after the application of GNS. The left panel shows the original KGE-EML model, where it is evident that the original KGE-EML model has clear category boundaries. However, due to the absence of a negative sampling strategy, a few scattered data points from other categories are still present.

In the right panel, we observe that KGE-EML+GNS not only increases the separation between data points from different categories but also addresses the issue of certain data points that could not be distinguished by the original KGE-EML model. This demonstrates that the proposed negative sampling and auxiliary loss functions in this paper effectively preserve the semantic information of entities in the knowledge graph and enhance their representational capability.



**Figure 7.** Visualizations before and after applying GNS: (a) the visualization results of KGE-EML embeddings; (b) the results after incorporating GNS negative sampling and auxiliary loss.

**Table 4.** Link prediction results with GNS using other negative sampling methods for different scoring functions.

| Models   | Methods   | FB15k-237  |              |              | WN18RR      |              |              |
|----------|-----------|------------|--------------|--------------|-------------|--------------|--------------|
|          |           | MR         | MRR          | Hit@10       | MR          | MRR          | Hit@10       |
| TransE   | Uniform   | 357        | 0.294        | 0.465        | 3384        | <b>0.226</b> | 0.501        |
|          | KBGAN     | 722        | 0.293        | 0.466        | 5356        | 0.181        | 0.432        |
|          | NSCaching | -          | 0.299        | 0.476        | 4472        | 0.2          | 0.478        |
|          | EANS      | 169        | 0.338        | 0.526        | 3488        | 0.222        | 0.526        |
|          | GNS       | <b>159</b> | <b>0.343</b> | <b>0.53</b>  | <b>2906</b> | 0.030        | <b>0.523</b> |
| TransD   | Uniform   | 188        | 0.245        | 0.429        | 3555        | 0.19         | 0.464        |
|          | KBGAN     | 825        | 0.247        | 0.444        | 4083        | 0.188        | 0.464        |
|          | NSCaching | 189        | 0.286        | 0.479        | <b>3104</b> | 0.201        | 0.484        |
|          | EANS      | 208        | 0.334        | 0.519        | 6937        | 0.218        | 0.476        |
|          | GNS       | <b>167</b> | <b>0.341</b> | <b>0.529</b> | 5199        | <b>0.226</b> | <b>0.495</b> |
| DistMult | Uniform   | 254        | 0.241        | 0.419        | 5110        | 0.43         | 0.49         |
|          | KBGAN     | 276        | 0.227        | 0.4          | 11351       | 0.204        | 0.295        |
|          | NSCaching | 273        | 0.283        | 0.456        | 7708        | 0.413        | 0.455        |
|          | EANS      | 397        | <b>0.309</b> | <b>0.482</b> | 4938        | <b>0.438</b> | 0.538        |
|          | GNS       | <b>176</b> | 0.302        | 0.477        | <b>3580</b> | 0.391        | <b>0.539</b> |
| ComplEx  | Uniform   | 339        | 0.247        | 0.428        | 5261        | 0.44         | 0.51         |
|          | KBGAN     | 881        | 0.191        | 0.321        | 7528        | 0.318        | 0.355        |
|          | NSCaching | 221        | 0.302        | 0.481        | 5365        | 0.446        | 0.509        |
|          | EANS      | 454        | <b>0.323</b> | <b>0.503</b> | 5350        | 0.463        | 0.558        |
|          | GNS       | <b>203</b> | 0.27         | 0.428        | <b>3842</b> | <b>0.468</b> | <b>0.559</b> |
| RotatE   | Uniform   | 187        | 0.295        | 0.478        | 3274        | 0.473        | 0.565        |
|          | EANS      | 169        | <b>0.341</b> | <b>0.528</b> | 3149        | 0.487        | 0.574        |
|          | GNS       | <b>162</b> | 0.334        | 0.52         | <b>3105</b> | <b>0.492</b> | <b>0.577</b> |

The bold text here in the “Model” column represents our proposed model, while under each evaluation metric, it signifies the best results.

## 5. Conclusions

The introduction of knowledge graphs has offered novel perspectives for knowledge representation and application. However, existing models face challenges such as incomplete structural information and inadequate semantic details in learned knowledge graph embeddings. This study addresses these issues through the following approaches.

Firstly, an investigation into the impact of the number of negative samples and different sampling strategies is conducted. The random sampling method is refined, introducing the Gibbs negative sampling method. Additionally, a contrast loss function is applied to mitigate gradient vanishing during training, expanding the influence of high-quality negative samples through the GNS module.

Secondly, to bolster relationship modeling, a metric learning-based KGE model is introduced. Entities' projection onto hyperplanes, determined by relations, is utilized. The semantic similarity measurement replaces the Euclidean distance comparison, enhancing the model's representational power. Furthermore, a graph attention network layer is incorporated to fuse embedded relation information, augmenting the expressive capabilities of relations. Link prediction experiments on public and private datasets demonstrated the superiority of the proposed KGE-EML model over baseline models, showcasing a high prediction accuracy and high generalization capabilities. Refinement tests based on relationship types illustrate the model's effectiveness in capturing complex relationships.

Finally, an auxiliary loss is introduced to extend the model into a generative adversarial network-based model. The experimental results indicate improved performance in link prediction compared to the original model. Furthermore, comparisons with other negative sampling methods affirm the significant advantage of the proposed GNS module.

The current study has certain limitations, particularly in its handling of one-to-one relationships, where the effectiveness does not reach its full potential. This could be attributed to the influence of high-quality negative sampling on processing capability. Nonetheless, it is worth noting that the current performance demonstrates significant improvements. Moving forward, future research will concentrate on refining the sampling methodology to address this limitation more effectively.

**Author Contributions:** Conceptualization, S.H.; Methodology, S.H. and M.H.; Writing—Original Draft, S.H.; Writing—Review & Editing, L.D., W.J. and Y.Q.; Software, M.H.; Supervision, L.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by The National Social Science Fund of China (Grant No. 23VRC094); The National Key Research and Development Program of China; 2021YFB3101300, 2021YFB3101302, 2021YFB3101305; Major Program of the National Social Science Foundation of China (Grant No. 22&ZD147); The National Natural Science Foundation of China under Grant: 62176009.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: <https://everest.hds.utc.fr/doku.php?id=en:transe>.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhang, T.; Tian, X.; Sun, X.; Yu, M.; Sun, Y.; Yu, G. Survey of knowledge graph embedding technology. *J. Softw.* **2023**, *34*, 277–311.
2. Bajwa, I.S. Context based meaning extraction by means of markov logic. *Int. J. Comput. Theory Eng.* **2010**, *2*, 35. [[CrossRef](#)]
3. Horita, K.; Kimura, F.; Maeda, A. Automatic Keyword Extraction for Wikification of East Asian Language Documents. *Int. J. Comput. Theory Eng.* **2016**, *8*, 32–35. [[CrossRef](#)]
4. Chatvichienchai, S. SEMEXSS—A Rule-Based Semantic Metadata Extraction System for Spreadsheets. *Int. J. Comput. Theory Eng.* **2016**, *8*, 102. [[CrossRef](#)]
5. Seneviratne, M.D.S.; Ranasinghe, D.N. Use of Agent Technology in Relation Extraction for Ontology Construction. *Int. J. Comput. Theory Eng.* **2012**, *4*, 884. [[CrossRef](#)]
6. Guo, S.; Wang, Q.; Wang, B.; Wang, L.; Guo, L. Semantically smooth knowledge graph embedding. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; pp. 84–94.
7. Kaya, M.; Bilge, H.Ş. Deep metric learning: A survey. *Symmetry* **2019**, *11*, 1066. [[CrossRef](#)]
8. Hoffer, E.; Ailon, N. Deep metric learning using triplet network. In Proceedings of the Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, 12–14 October 2015; Proceedings 3; Springer: Berlin/Heidelberg, Germany, 2015; pp. 84–92.

9. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28, pp. 1112–1119.
10. Chao, L.; He, J.; Wang, T.; Chu, W. Pairre: Knowledge graph embeddings via paired relation vectors. *arXiv* **2020**, arXiv:2011.03798.
11. He, M.; Duan, L.; Zhang, B.; Han, S. Knowledge Graph Embedding Method Based on Entity Metric Learning. In Proceedings of the 2023 9th International Conference on Computing and Artificial Intelligence, Tianjin, China, 17–20 March 2023; pp. 381–387.
12. Geng, Y.; Chen, J.; Chen, Z.; Pan, J.Z.; Ye, Z.; Yuan, Z.; Jia, Y.; Chen, H. OntoZSL: Ontology-enhanced Zero-shot Learning. In Proceedings of the Web Conference, Ljubljana, Slovenia, 19–23 April 2021; pp. 3325–3336.
13. Zhang, N.; Deng, S.; Sun, Z.; Chen, J.; Zhang, W.; Chen, H. Relation adversarial network for low resource knowledge graph completion. In Proceedings of the Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 1–12.
14. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for knowledge graph completion. *arXiv* **2019**, arXiv:1909.03193.
15. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
16. Yang, Z.; Ding, M.; Zhou, C.; Yang, H.; Zhou, J.; Tang, J. Understanding negative sampling in graph representation learning. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 1666–1676.
17. Cai, L.; Wang, W.Y. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv* **2017**, arXiv:1711.04071.
18. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
19. Wang, P.; Li, S.; Pan, R. Incorporating GAN for Negative Sampling in Knowledge Representation Learning **2018**. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 2005–2012.
20. Xu, D.; Xu, T.; Wu, S.; Zhou, J.; Chen, E. Relation-Enhanced Negative Sampling for Multimodal Knowledge Graph Completion. In Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Lisboa, Portugal, 10–14 October 2022; pp. 3857–3866. [[CrossRef](#)]
21. Zhang, Y.; Yao, Q.; Chen, L. Simple and automated negative sampling for knowledge graph embedding. *VLDB J.* **2021**, *30*, 259–285. [[CrossRef](#)]
22. 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*.
23. 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.
24. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the 28th International Conference on Machine Learning (ICML), Bellevue, WA, USA, 28 June–2 July 2011; Volume 11, pp. 3104482–3104584.
25. Yang, B.; Yih, W.t.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv* **2014**, arXiv:1412.6575.
26. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 19–24 June 2016; pp. 2071–2080.
27. Balažević, I.; Allen, C.; Hospedales, T.M. Tucker: Tensor factorization for knowledge graph completion. *arXiv* **2019**, arXiv:1901.09590.
28. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 1811–1818.
29. Vashishth, S.; Sanyal, S.; Nitin, V.; Agrawal, N.; Talukdar, P. Interact: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3009–3016.
30. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and deep locally connected networks on graphs. *arXiv* **2014**, arXiv:1312.6203.
31. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; IEEE: New York, NY, USA, 2005; Volume 1, pp. 539–546.
32. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
33. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015; Volume 2.
34. Chen, T.; Kornblith, S.; Norouzi, M. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 1597–1607.
35. Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y. Learning structured embeddings of knowledge bases. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–11 August 2011; Volume 25, pp. 301–306.
36. Xie, R.; Liu, Z.; Sun, M. Representation learning of knowledge graphs with hierarchical types. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), New York, NY, USA, 9–15 July 2016; Volume 2016, pp. 2965–2971.
37. Liu, X.; Tan, H.; Chen, Q.; Lin, G. RAGAT: Relation aware graph attention network for knowledge graph completion. *IEEE Access* **2021**, *9*, 20840–20849. [[CrossRef](#)]



38. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018; Proceedings 15; Springer: Berlin/Heidelberg, Germany, 2018; pp. 593–607.
39. Degraeve, V.; Vandewiele, G.; Ongenaes, F.; Van Hoecke, S. R-GCN: The R could stand for random. *arXiv* **2022**, arXiv:2203.02424.
40. Je, S.H. Entity Aware Negative Sampling with Auxiliary Loss of False Negative Prediction for Knowledge Graph Embedding. *arXiv* **2022**, arXiv:2210.06242.

**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.