



**HAL**  
open science

# Knowledge graph representation learning: a comprehensive and experimental overview

Dorssaf Sellami, Wissem Inoubli, Imed Riadh Farah, Sabeur Aridhi

## ► To cite this version:

Dorssaf Sellami, Wissem Inoubli, Imed Riadh Farah, Sabeur Aridhi. Knowledge graph representation learning: a comprehensive and experimental overview. *Computer Science Review*, 2024, 56. <hal-04853146>

**HAL Id: hal-04853146**

**<https://hal.science/hal-04853146v1>**

Submitted on 2 Jan 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Knowledge graph representation learning: a comprehensive and experimental overview

Dorsaf Sellami<sup>a</sup>, Wissem Inoubli<sup>b</sup>, Imed Riadh Farah<sup>a</sup>, Sabeur Aridhi<sup>c</sup>

<sup>a</sup>*Manouba university, ENSI, RIADI, 2010, Manouba, Tunis*

<sup>b</sup>*Artois University, CNRS, UMR, 8188, CRIL, Lens, France*

<sup>c</sup>*Université de Lorraine, CNRS, Inria Nancy Grand-Est, LORIA, 54000, Nancy, France*

---

## Abstract

Knowledge graph embedding (KGE) is a hot topic in the field of Knowledge graphs (KG). It aims to transform KG entities and relations into vector representations, facilitating their manipulation in various application tasks and real-world scenarios. So far, numerous models have been developed in KGE to perform KG embedding. However, several challenges must be addressed when designing effective KGE models. The most discussed challenges in the literature include scalability (KGs contain millions of entities and relations), incompleteness (missing links), the complexity of relations (symmetries, inversion, composition, etc.), and the sparsity of some entities and relations. The purpose of this paper is to provide a comprehensive overview of KGE models. We begin with a theoretical analysis and comparison of the existing methods proposed so far for generating KGE, which we have classified into four categories. We then conducted experiments using four benchmark datasets to compare the efficacy, efficiency, inductiveness, the electricity and the CO2 emission of five state-of-the-art methods in the link prediction task, providing a comprehensive analysis of the most commonly used benchmarks in the literature.

*Keywords:* Knowledge graphs, knowledge graph embedding, representation space, link prediction, scalability.

*PACS:* 0000, 1111

*2000 MSC:* 0000, 1111

---

## 1. Introduction

In recent years, knowledge graphs (KGs) have become widely used in the field of artificial intelligence. Their ability to store and represent factual knowledge has led to successful applications in multiple domains, including information retrieval [1], natural language processing [2], intelligent question answering systems [3], recommendation systems [4], and more.

The concept of KGs was first introduced by Google in 2012 [5], defined as a large-scale knowledge base composed of a large number of entities and the relations between them. Alongside big data and deep learning, KGs are now one of the main driving forces behind the development of artificial intelligence [6]. Although Freebase [7], DBpedia [8], and YAGO [9] are widely used for large-scale KGs that include millions of entities and relations, many missing facts and implicit relations between entities remain unexplored, leading to incomplete structures and content in KGs [10].

The expert system developed in the late 1960s [11] is a precursor to KGs, which are considered intelligent systems that incorporate knowledge and data on a large scale. By combining knowledge and data, researchers aim to build knowledge-based systems. In the early stages of modern KG construction, it was common to codify the knowledge of experts in a particular field, such as semantic web and KG classification.

In the literature, KGs are described as multi-relational graphs composed of entities (or nodes) and relations (or edges). A KG is represented as a set of triples, also called facts, which are denoted in the form  $(h, r, t)$ , where  $h$  represents the head entity,  $r$  represents the relation, and  $t$  represents the tail entity, indicating that two entities are connected by a specific relation [12]. Currently, the main focus of KG research is on three aspects: knowledge representation [13], KG construction [14], and KG application [15]. These aspects combine cognitive computing, knowledge representation and reasoning, information retrieval and extraction, natural language processing, data mining, and other technologies [6].

Although KGs represent structured factual information effectively, but their large-scale and complicated graph structure make them difficult to manipulate [16]. Entities have intricate and complex relations that involve structures such as rings and hierarchies. [16]. Additionally, the complexity of relations which can be assessed based on criteria such as (1) the types of relations (e.g., symmetric/antisymmetric, transductive, inverse, and hierarchical relations). (2) The cardinality (the number of entities involved in a

relationship, such as one-to-one or one-to-many), and (3) the interconnect-  
edness (the degree to which relations are interrelated or dependent on each  
other) requires embeddings that are expressive enough to capture these nu-  
ances. Certain entities and relations are also rare [90], presenting further  
challenges. Consequently, scalability, incompleteness, complex real-world re-  
lations, and sparsity are the most challenging aspects of KGs. Therefore, it  
is crucial to effectively and efficiently extract and leverage useful information  
from large-scale KGs for downstream machine learning (ML) tasks, such as  
node classification [17], community detection [18], and link prediction [19].

To facilitate the learning of KGs in ML downstream tasks, KGE, also  
known as Knowledge Representation Learning (KRL), was proposed and  
has received significant attention within the ML community. The essential  
idea of KGE is to embed entities and relations in a low-dimensional space  
while preserving as much of the rich information in KGs as possible [20].  
KGE models, using optimization strategies, can generate embeddings (vector  
representations) that capture the latent properties of entities and relations in  
the graph [21]. The latent properties of entities and relations refer to hidden  
characteristics that can be inferred through modeling techniques, capturing  
essential information about entity interactions and the semantics of relations.  
These properties are typically represented as latent features and help evaluate  
the complexity of relations, which includes cardinality types (one-to-one, one-  
to-many, many-to-many) and multi-hop connections that require multiple  
intermediate steps for interpretation.

The purpose of this paper is to provide an overview of the latest lit-  
erature on KGE methods, which enhances graphs with additional context,  
intelligence, semantics for knowledge acquisition and knowledge-aware appli-  
cations. The contributions of this paper are summarized as follows:

- We present a detailed examination of current KGE methods, which we  
categorize into four groups. Following this, we conduct an in-depth the-  
oretical analysis of KGE methods based on key metrics such as types  
of relations (e.g., symmetric/asymmetric, compositional, or inverse),  
training approach (inductive or transductive), and computational com-  
plexity.
- We include an overview of real-world applications, commonly used  
datasets, and popular application tasks, along with their evaluation  
metrics, which are currently used for benchmarking. Additionally, we  
highlight widely used open-source libraries in KGE research.

- We conduct a comparative analysis to assess the effectiveness, efficiency, and inductive capacity of selected KGE models. The experimental methodology centers on using link prediction as a downstream task to evaluate these models. This comparison involves an examination of each model’s scoring functions, training procedures, and generalizability across various datasets. The results demonstrate each model’s strengths and limitations in predicting missing links, computational efficiency, and inductive capabilities across different scenarios. Furthermore, we assess resource consumption, including CPU, GPU, and RAM energy and power usage, total electricity consumption, and estimated CO2 emissions.

The remainder of this paper is structured as follows. In Section 2, we begin by introducing existing KGE models. We also discuss some real-world applications, the most used datasets, and the main popular metrics in each application task in Section 3. In Section 4, we present the experimental results of these models and a discussion. In Section 5, we present the most open challenges and future work. Finally, in Section 6, we make concluding remarks.

## 2. Related works

In the literature, there is a wide range of KGE methods, which are classified into different categories. Initially, in [26] KGE methods were classified into two main categories: distance-based models and semantic matching models. For other categorization, we have [23] [19] [5] [27], KGE methods are classified into three main categories: translation-based methods, semantic matching-based methods, and neural network-based methods. Furthermore, Ji et al. [25] classify KGE methods into four main families: KGE, knowledge acquisition, temporal KGs, and knowledge-aware applications.

Table 1 presents a summary of various research studies focusing on KG tasks, datasets, metrics, and key findings. The primary task across most studies is link prediction, with one study additionally covering Triple classification. The datasets frequently used include FB15k, WN18, and their variations such as FB15k-237 and WN18RR, as well as other datasets like FB13 and YAGO3-10. Metrics used in these studies include Mean Rank (MR), Hits@k (for k=1, 3, 10), Mean Reciprocal Rank (MRR), Accuracy, and Area Under the Precision-Recall Curve (AUC-PR). Discussions in these

studies typically address challenges in knowledge graph completion [91], including data sparsity and incompleteness. The studies explore various KGE Methods, presenting theoretical analyses, experimental results, and efficiency comparisons. Additionally, some studies analyze the inductive and transductive properties of neural network-based models, providing insights into the strengths and limitations of different KGE methods.

Table 1: Summary of related surveys

Ref	Tasks	Data	Metrics	Findings
[23]	Link prediction	FB15k WN18	MR Hits@10	<ul style="list-style-type: none"> <li>- Discuss how data sparsity affects large-scale KG systems</li> <li>- A description of KGE methods</li> <li>- The experimentation results are collected from different papers</li> </ul>
	Triple classification	WN11 FB13	Accuracy	
[27]	Link prediction	FB15k WN18 FB15k-23 WN18RR YAGO3-10	Hits@1 Hits@10 MR MRR	<ul style="list-style-type: none"> <li>- Discuss the problem of incompleteness in KG.</li> <li>- Presentation of KGE models</li> <li>- Analyze the results for efficiency and effectiveness.</li> </ul>
[19]	Link prediction	FB15k-237 FB15k	Hits@1 Hits@3 Hits@10 MRR	<ul style="list-style-type: none"> <li>- Discuss the problem of incompleteness in KG.</li> <li>- Theoretical analysis and comparison of the current KGE methods</li> <li>- The experimentation results are collected from different papers</li> </ul>
[105]	Link prediction	WN18R FB15k-237	AUC-PR	A comparison between the inductive and transductive properties of neural networks-based models.
[106]	Link prediction	WN18RR FB15k-237	Hits@10 AUC-PR	

### 3. Taxonomy of knowledge graph embedding techniques

KGE aims at embedding and representing entities and relations in a low-dimensional continuous vector space. This space is designed to preserve as much graph structure and property information as possible, while also facilitating calculations involving entities and relations. Most current methods use the Triples stored in the KG to perform the embedding task, ensuring that the embeddings are compatible with these Triples.

The literature presents numerous KGE methods, which can be categorized into four main groups: translational-based methods, tensor factorization-based methods (also known as semantic matching-based methods), random-walk-based methods, and neural network-based methods.

### 3.1. Random walk based methods

Many recent methods that belong to the class of direct encoding approaches [39] learn node embeddings based on random walk statistics [36]. Random Walk (RW) aims to generate random walks for each node in the graph to capture the graph structure and output similar node embedding vectors for nodes that occur in the same random walks. A RW describes a path consisting of a succession of random steps. This strategy includes various methods that we categorize into the type of graph: homogeneous graph [37] and heterogeneous graph [37]. The first type of graph is a graph with a single type of node and single relation for example a social network. The second type is graphs that have multiple types of connections, multiple types of nodes, multiple types of links between them and this is the notion of heterogeneous graphs such as heterogeneous information network (HIN)[37] and KG.

#### 3.1.1. Homogeneous graphs

In the first category of graphs, we present the most well-known methods for random walking on this type of graph. Deep walk [40] and node2vec [41] are two methods that use random walk with skipgram [42]. Deep walk method differ in that they generate random walks starting at each node in the graph to create sentences. Using the SkipGram algorithm, node embeddings are generated for these sentences. But Node2Vec can be viewed as a generic version of DeepWalk, where Node2Vec’s walks are random, but biased by two hyperparameters,  $p$  and  $q$  [38]. When creating walks, these parameters are employed to increase the likelihood that the path returns to its parent node or is further away from it.

We can find, also, Large-Scale Information Network Embeddings (LINE) algorithm [43] where it does not explicitly use random walks, it shares the same conceptual motivations as DeepWalk and Node2vec. In LINE, the main idea is to combine two encoders and two decoders. The first objective aims to encode first-order adjacency information while the second objective has a more similarity to random walk approaches.

#### 3.1.2. Heterogeneous graphs

For heterogeneous graphs, numerous methods have been proposed. Firstly, the methods used for HIN where we find Yuxiao Dong et al. [45] which proposes a metapath2vec model. This model is for HIN which learns desirable node representations where entities of different types are interconnected by

various types of relations. The idea behind this method is to maximize probability in consideration of multiple types of node entities and relations. The `metapath2vec` model utilizes meta-path-based random walks to create a heterogeneous neighborhood for each node and then employs a heterogeneous skip-gram model to generate node embeddings. Another method we can mention is HIN2Vec [44], which captures the rich semantics of HINs by using various types of relations between nodes. HIN2Vec jointly performs multiple prediction training tasks to learn the latent vectors of nodes and metapaths in the HIN when given a set of relationships defined as metapaths in the HIN (which reduces the prediction tasks of the conceptual NN model into new prediction tasks).

Secondly, we have the following KG methods presented by researchs. Eneko Agirre et al. [46] present a method for knowledge-based Word Sense Disambiguation (WSD) based on random walks over relations in a large lexical knowledge base. Zhang Hui et al [47] propose a fast training framework for KGR based on RW strategy based on relationel path and relationel encodding (RWRRel) which unites relation-specific subject embeddings with relation-specific object embeddings to model the segments of relations in KG.

Due to the fact that RW cannot obtain the syntactic meaning behind the relations in the KG, embedding is considered the syntactic embedding of the subject (head) and object relation (tail). This strategy aids in fast training of KG by RW in relation paths.

Ali Assia and Wajdi Dhifli [48] propose Semantic and Bipartite Graph-Based Instance Matching (SBIGMat) is an approach to the Instance Matching Problem based on Markov RW. This approach leverages both local and global information, which is mutually calculated from a pairwise similarity graph.

SBIGMat can remain fast and scalable by capturing semantic similarity between source instances and candidate targets from equally ranked nodes with semantic post-processing. This allows for better instance-matching results using semantic post-processing.

Qiao Liu et al. [49] propose a hierarchical random walk inference algorithm (HiRi) for relational learning in large-scale graph-structured knowledge bases, which maintains the computational simplicity of random walk models and provides better inference accuracy.

To be precise, it explains a two-tier random-walk mechanism for retrieving relations. The model’s upper tier is associated with the process of recognising and learning relation sequence patterns from a global perspective. The lower

tier is to obtain valuable information from the relation-specific subgraphs in GKBs, as each subgraph only represents one particular type of relationship.

### 3.2. Fact-based representation learning methods

The facts observed in KG are stored as a collection of triples. Each triple  $(h,r,t)$  is composed of a head entity ( $h$ ), a tail entity ( $t$ ), and a relation ( $r$ ) between them. KGE consists of embedding entities and relations in a low-dimensional continuous vector space. The embedding task is usually performed using facts stored in the KG, which require embedding to be compatible with the facts. Three steps are typically included in typical KGE methods: the initial step involves representing entities and relations, which defines the manner in which they are represented using a pointwise space [28] that includes the tensor, vector and matrix. Entities are typically represented as vectors and relations are typically treated as operations in the vector space, which can be represented as tensors, vectors, and matrices, while other types of space such as Gaussian space [54], complex vector space [34] and manifold space [50].

The second step aims to compute the scoring function [51]. When a scoring function  $f_r(h,t)$  is defined for each fact  $(h,r,t)$  to measure its plausibility. The KG tends to assign higher scores to facts that have been observed compared to those that have not been observed. The last step is to learn the representations of entities and relations, and to solve an optimization problem that maximizes the total plausibility of the observed facts.

In recent years, translational and tensor factorization-based methods (semantic matching methods) have become highly popular because they are simple and can work with the open-world assumption (OWA) [99] in KG.

#### 3.2.1. Translation based methods

Translation-based methods or distance model-based scoring function [52] in KGC and link prediction is one of the most popular strategies that use KGE methods to measure the plausibility of facts by calculating the euclidean distance between entities. The intuition behind this strategy is to learn embeddings by representing a relation as a translation from  $h$  to  $t$ , with the objective of minimizing the distance between  $h$  and  $t$  after applying the relational transformation.

To achieve an effective embedding method capable of discerning the authenticity of Triples, TransE (Translating Embeddings) [28] has emerged as one of the most popular models based on KGE scoring functions. This model

is introduced to represent entities and relations as translations in the embedding space that is presented in Figure 1 a. TransE is specifically aimed at minimizing the distance between  $h$  and  $t$  when connected by a known relation, where the goal is to predict missing relationships between entities. The score function of TransE is defined as a distance between  $h + r$  and  $t$  under  $l_1$ -norm or  $l_2$ -norm constraints. In mathematical expressions, it is shown as follows:

$$f_r(h, t) = -\|h + r - t\|_{L_1/L_2} \quad (1)$$

Despite its significant progress in large-scale KGE, TransE encounters difficulties when dealing with complex relations where an entity can have multiple types of relationships. These include one-to-many (1-n) relations, where a single entity is linked to several entities (e.g., a teacher who teaches multiple students); many-to-one (n-1) relations, where multiple entities are connected to a single entity (e.g., several authors writing for the same journal); and many-to-many (n-n) relations, where multiple entities are related to multiple others (e.g., students enrolling in various courses). To address these challenges, numerous variants and extensions of TransE have been proposed, enhancing its ability to handle such complex relational structures.

TransH (Translating Embeddings in a Hyperplane) [29] extended the original TransE model where each entity has different embedding space when the entity is involved in diverse relations and allows each relation to hold its own relation-specific hyperplane such as presented in Figure 1 b. The idea of TransH is to project entities into hyperplanes associated with their relations. This model is designed to capture the diverse semantics of different relations, making it suitable for KG with diverse relations. The score function of this model is presented as follows where  $w_r$  be the normal vector to a relation-specific hyperplane

$$f_r(h, t) = -\|(\mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r) + \mathbf{r} - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r)\|_2^2 \quad (2)$$

TransH enables each entity to have a different representation corresponding to its different relations, but the entities and relations are still represented in the same feature space. An entity might contain various semantic meanings, while a relation might be able to capture entities' diverse aspects; hence, modeling entities and relations in the same semantic space could be inadequate for graph embedding.

Other models were developed to address the limitations of TransE, particularly its difficulty in modeling complex relations with varying semantics.

TransR (Translating Embeddings for Relationships) [30] builds upon the TransE model by extending the concept of relation-specific hyperplanes introduced by TransH to relation spaces. In an entity vector space,  $h$  and  $t$  are the representations of each Triples  $(h, r, t)$ , and the relations are represented by a translation vector  $r$  in a relation-specific space. The scoring function of this model as presented as follows where  $M_r$  present a projection matrix to transform thr entity vectors into the relation-specific space:

$$f_r(h, t) = - \|M_r h + r - M_r t\|_2^2 \quad (3)$$

In particular, TransR creates a projection matrix for each relation  $r$  to transform the entity vectors into the relation space as presented in Figure 1 c. Nonetheless there are some drawbacks, such as the fact that the  $h$  and  $t$  entities have the same projection matrix for a relation, it is understandable that there may be fundamental differences in their types or attributes. In addition, entities and relations are involved in an interactive process when projecting from the entity space to the relation space. Integrated information cannot be captured, also, when the projection matrix is only related to relations. Finally, due to the use of the projection matrix, TransR requires a large amount of computing resources. To address these shortcomings, TransD [31] was proposed as an optimization of TransR. It uses two vectors for each entity-relation pair to construct a dynamic mapping matrix, serving as a substitute for the projection matrix presented in Figure 1 d. TransD actually introduces the concept of translation matrices that are specific to relations. Each relation has its own translation matrix, which is used to transform the entity vectors in different ways for each relation. This allows TransD to capture more complex patterns in the data compared to TransE (such as the difficulty modeling the 1-N and N-1 relations). The score function of TransD is defined as follows:

$$f_r(h, t) = - \|(\mathbf{w}_r \mathbf{w}_h^\top + \mathbf{I}) \mathbf{h} + \mathbf{r} - (\mathbf{w}_r \mathbf{w}_t^\top + \mathbf{I}) \mathbf{t}\|_2^2 \quad (4)$$

Where  $w_h$ ,  $w_t$  and  $w_r$  are a mapping vectors along with the entity/relation presentations and  $I$  is an identity matrix.

The TransD model constructs a dynamic mapping matrix with two projection vectors, which effectively reduces computational complexity. To overcome some drawbacks of the indicated methods, such as the heterogeneity of the relations (which leads to underfitting in complex relations or overfitting in simple relations) and the imbalance (the quantities in the relation differ

between head and tail), TranSparse [53] was proposed. This model requires that the transformation matrix be sparse. The score function of this model is presented as follows:

$$f_r(h, t) = - \|\mathbf{M}_r(\theta_r)\mathbf{h} + \mathbf{r} - \mathbf{M}_r(\theta_r)\mathbf{t}\|_{1/2}^2 \quad (5)$$

where  $M_r(\theta_r)$  are a sparse matrices to replace dense projection matrices for each relation  $r$  and the sparse degree  $\theta_r$  is linked to the number of entities connected with relation  $r$ .

RotatE [58] introduces the rotational Hadamard product, treating the relation as a rotation between the  $h$  and  $t$  entities in a complex space. The score function is defined as follows:

$$f_r(h, t) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\| \quad (6)$$

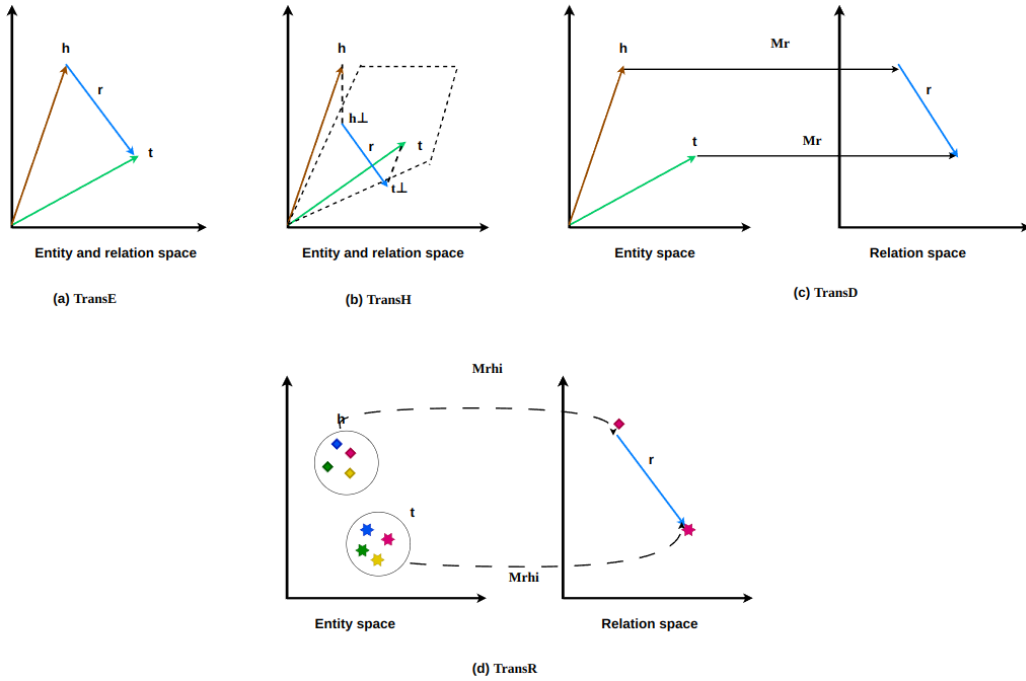


Figure 1: Illustration of such a translation methods

KG embedding Translation-based methods usually consider entities and relations as vectors embedded in low-dimensional semantic spaces. However,

the entities and relations in the KGs are various with different granularities. Therefore, the margin in the margin-based score function that is used to distinguish positive triples from negative triples should be more flexible due to diversity, and the uncertainties of entities and relations should be taken into account.

To address the above shortcoming, KG2E [54] was proposed, introducing multidimensional gaussian distributions to the KG representations [55]. KG2E represents each entity and relation with a random vector. The scoring function proposed in TransE is extended in KG2E to learn Gaussian distributions for entities and relations instead of fixed points, thereby capturing uncertainty and variability.

Another notable model is TransG [56], which is based on Gaussian distribution. TransG addresses the situation of multiple relation semantics, where a single relation associated with different entity pairs may encompass multiple meanings.

### 3.2.2. Tensor factorization-based methods

. Translation-based methods such as TransE and its extensions are simple but efficient for many types of relations, which can be used for various tasks such as KGC, triple classification, and link prediction. However, there are also some other embedding methods that are tensor factorization-based methods that perform well on KG representations. The main idea behind these methods is as follows: The process begins with transforming the facts represented in a KG into a three-dimensional binary tensor, which effectively structures the data for further analysis by capturing the relationships between entities and their attributes. Next, embedding matrices corresponding to the embedding vectors of both entities and relations are computed. These matrices play a crucial role in representing the tensor through a process known as factorization, simplifying the complex relationships inherent in the data. Finally, these embedding matrices are utilized to generate low-dimensional representations for each entity and relation. This dimensionality reduction is essential for making the data more manageable and suitable for various machine learning algorithms, ultimately enhancing both computational efficiency and performance in downstream tasks. By following these steps, we can effectively leverage KGs for advanced analytics and machine learning applications.

RESCAL [57] (see Figure 2 a) assigns a vector to every entity to capture its latent semantics. A matrix  $M_r$  is used to model pairwise interactions

between latent factors for each relation. The RESCAL scoring function can be defined as:

$$f_r(h, t) = h^T M_r t \quad (7)$$

DistMult [32] (see Figure 2 b) uses a bilinear model to simplify relations representation by assuming that interactions between entities are symmetric. Entities and relations are represented as vectors in this model, and the dot product is utilized to score the plausibility of triples. Although DistMult is easy to use, its symmetry assumption can cause problems with asymmetric relations. The DistMult scoring function can be written as:

$$f_r(h, t) = h^T \text{diag}_r t \quad (8)$$

where  $\text{diag}_r$  is the diagonal matrix constructed from  $r$ .

Holographic Embedding (HoLE) [33] (see Figure 2 c) merges RESCAL expressive power with DistMult efficiency and simplicity. It is a model that combines circular correlation with entity embeddings to capture rich interactions while still being efficient than RESCAL. HoLE’s use of holographic principles enables it to model both symmetric and asymmetric relations and capture complex patterns within the KG, while maintaining low computational requirements. The scoring function of this model can be written as:

$$f_r(h, t) = h^T (h * t) \quad (9)$$

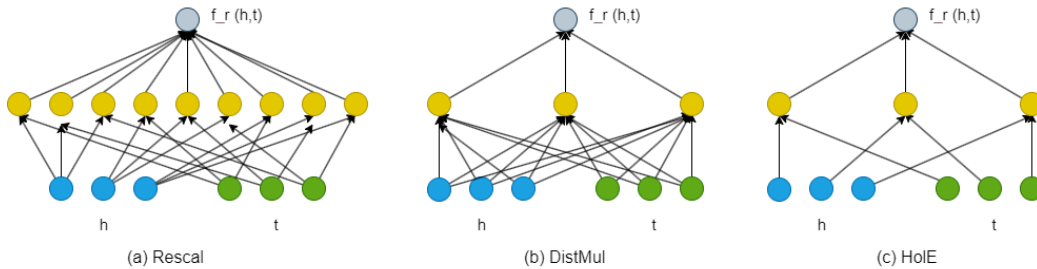


Figure 2: Illustration of such tensor factorization-based methods

Complex Embedding (ComplEx) [34] extends DistMult by introducing complex-valued embeddings to improve the model of asymmetric relations. The embeddings of the entity and the relation  $h, r$ , and  $t$  in ComplEx are

no longer located in a real space but in a complex space  $C^d$ . The ComplEx scoring function can be presented as:

$$f_r(h, t) = \text{Re}(h^T \text{diag}_r \tilde{t}) \quad (10)$$

where  $\tilde{t}$  denote the complex conjugate of the  $t$

ANALOGY [35] expands RESCAL in order to further model the analogical properties of entities and relations. To model analogical structures, it is necessary to have relations and linear maps that are normal and mutually commutative. ANALOGY uses matrices  $M_r$  to represent relations, but they can also be blocked diagonalized into a set of sparse almost-diagonal matrices that contain only  $O(d)$ . The ANALOGY scoring function can be defined as:

$$f_r(h, t) = h^T M_r t \quad (11)$$

Finally, the geometric algebra-based KG embedding framework (GeomE) [59] employs multivector representations and the geometric product to create models of entities and relationships. The model consists of multiple advanced KG embedding approaches and is advantageous in its capacity to model various key relation patterns, including anti-symmetry, inversion, and composition. It also exhibits rich expressiveness with a high degree of freedom and good generalizability. The score function of GeomE is defined as the scalar of the product of the embeddings of  $h$ ,  $r$  and  $t$  by using the geometric product and the Clifford conjugation:

$$\phi^{\text{GeomE}}(h, r, t) = \langle \text{Sc}(\mathbf{M}_h \otimes_n \mathbf{M}_r \otimes_n \overline{\mathbf{M}}_t), \mathbf{1} \rangle \quad (12)$$

where  $n = 2$  for GeomE2D and  $n = 3$  for GeomE3D,  $\otimes_n$  denotes element-wise Geometric Product between two  $k$  dimensional  $n$ -grade multivectors (e.g.  $M_h \otimes_n M_r = [M_{h1} \otimes_n M_{r1}, \dots, M_{hk} \otimes_n M_{rk}]$ ),  $Sc$  denotes the scalar component of a multivector,  $\mathbf{1}$  denotes a  $k$ 1 vector having all  $k$  elements equal to one,  $\overline{\mathbf{M}}$  denotes the element-wise conjugation of multivectors i.e.  $\overline{\mathbf{M}} = [\overline{\mathbf{M}}_1, \dots, \overline{\mathbf{M}}_k]$ .

### 3.3. Neural network based methods

In many different fields, deep learning is one of the most popular and widely used models. These models possess strong representation and generalization capabilities that enable them to express complex nonlinear projections. In recent years, the use of a neural network to embed a KG into a continuous feature space has become a hot topic. Semantic Matching Energy

model (SME) [60] uses the neural network architecture to achieve semantic matching, which aims to learn continuous vector representations for entities and relations that capture semantic relations and similarities within the KG. The SME can be defined as the distance or energy between the embeddings of entities involved in a relations where the goal is to minimize this energy for correct triples (h, r, t) and maximize it for incorrect ones. The scoring function is defined with two versions of matching blocks: linear and bilinear block:

$$f_r(h, t) = g_l(h, r)^T g_r(r, t) \quad (13)$$

Where  $g_l(h, r)$  the combination of  $r$  with  $h$  and  $g_r(r, t)$  the combination of  $r$  with  $h$ .

The neural tensor network (NTN) [61] was introduced to address the limitations of simpler models (i.e TransE, Rescal, HolE, etc.) in capturing nonlinear relations and representing the rich structure of KG. The scoring function of this model is:

$$f_r(h, t) = r^T \tanh(h^T M_r t + M_r^1 t + M_r^2 t + b_r) \quad (14)$$

where  $M_r$  is the combination of  $handt$  by a relation -specific tensor and  $M_r^1, M_r^2$  and  $b_r$  are relation-specific weight matrices and bias vectors.

Neural Association Model (NAM) [62] is a nonlinear model for probabilistic reasoning which aims to use neural networks to model the association probability for any two events. The scoring function of this model is:

$$f_r(h, t) = t^T z^{(l)} \quad (15)$$

$z^{(l)} = [h; r]$  the vector embeddings of h and r in the input layer.

In recent years, due to their great ability to represent graph structures, graph neural networks (GNNs) have become more popular. Among recent methods based on GNN [63] we mentioned Relational Graph Convolutional Network (R-GCN) [64] for heterogeneous graphs such as KG which is an application of the GCN [65] framework which applies a nonlinear multilayer convolution model to graph. R-GCN can capture the rich relational structure of KG. By aggregating information from its neighbors and considering the types of relations connecting them, it updates the embedding of each entity. This makes it possible for RGCNs to capture complex relations patterns and interactions in the graph. The forward propagation is formulated as follows:

$$x_i^{(l+1)} = \sigma \left( \sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} x_j^{(l)} + W_0^{(l)} x_i^{(l)} \right) \quad (16)$$

where  $x_i^{(l)}$  signifies the hidden state of the entity  $i$  in  $l$ -th layer,  $N_i^r$  indicates a neighbor collection where it connects to entity  $i$  with  $r$  and  $W_o^{(l)}$  are the weight matrices, and  $c_{i,r}$  denotes the normalization process. The purpose of Knowledge Graph Attention Network (KGAT) [66] is to create neural network architecture for recommendation systems that utilize KGs. The model is meant to tackle issues in recommendation scenarios where a bipartite graph is a natural representation of user-item interactions, and a KG contains additional semantic information. KGAT improves entity and relations representation by concentrating on the most relevant parts of the KG, which leads to more precise and context-aware embeddings. The scoring function of this model is:

$$f(h, r, t) = \|W_r \mathbf{e}_h + \mathbf{e}_r - W_r \mathbf{e}_t\|_2^2 \quad (17)$$

where  $W_r$  is the transformation matrix of  $r$ , which projects entities from the  $d$  dimension entity space into the  $k$  dimension relation space.  $e_h$ ,  $e_t$  and  $e_r$  are the embedding for  $h$ ,  $t$ , and  $r$ .

ConvE [67] is a convolutional network that uses 2D convolutional and multiple non-linear features to model KG, which involves reshaping  $h$  and  $t$  into 2D matrix. It was introduced to capture complex relations between entities in KGs by capturing local patterns and features, leading to more expressive and accurate embeddings and enhancing the ability to reason and infer over large and complex KGs. The scoring function of this model is:

$$f_r(h, t) = f(\text{vec}(M(h, r)) \mathbf{W}) \mathbf{t} \quad (18)$$

where  $(M(h, r))$  are the 2D reshapings of the head embedding and relation embedding and  $w$  is a matrix which is the responsible for the linear transformation that vectorizes and projects the tensor into  $k$ -dimensional space.

### 3.4. Summary

KGE is crucial to the research community that is focused on KG. This section reviews four folds of KGE with several modern methods summarized in Figure 4 which represents the methods with their taxonomy. Table 3 represents a comparison between methods depending the type of relation (symmetric (Sym) / antisymmetric (Antisym), the composition and the inversion), the type of training (inductive (Ind) or transductive (Trans)), the types of learning (supervised (Super), semi-supervised (Semi) or unsupervised (Unsuper)) and the temporal complexity (Comlx) of each models. Specific notations and their descriptions are listed in Table 2.

Table 2: Notations and descriptions

Notation	Description
h	Head entity
t	Tail entity
r	Relation
V	Number of nodes
K	Number of walks
N	Number of edges
d	Dimensional of entity
k	Dimensional of relation
l	Number of hidden layers in the network
G	All the graph

- Symmetric [68]: The relation between two entities holds true in both direction (Anna Friend of Lisa):  $r(h, t) = r(t, h)$  where h represent the head, t represent the tail and r represent the relation between two entities like presented in Figure 3 (a).
- Anti-symmetric [68]: The relation between two entities holds true in one direction, but not in the opposite direction( hypernym):  $r(h, t) = \neg r(t, h)$  like presented in Figure 3 (b).
- Composition [70]: The relation holds true if there is a chain of intermediate entities that connect them (my mother’s husband is my father):  $r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z)$  like presented in Figure 3 (c).
- Inversion [70]: The relation holds true in both directions, but with the head and tail entities reversed (Advisor, advisee):  $h + r_2 = t \quad \& \quad t - r_1 = h \Rightarrow r_1 = -r_2$  like presented in Figure 3 (d).
- Transductive [71]: The input graph can be observed over all the datasets splits (the dataset consists of one connected graph). The entire graph can be observed in all dataset splits, we only split the nodes (some labels are observed other labels are unobserved). Essentially, we will utilize the same graph structure for training, validation, and test set, and we will retain the same graph structure. We will only split node labels, which means we will keep the graph structure as is, but we

will add some nodes to the training, validation, and test sets. During training, we calculate the embeddings of nodes like A, B, and C, as demonstrated in Figure 3 (e), as they form the training set that utilizes the entire graph but only includes the nodes A, B, and C. During validation, we compute the embeddings for the entire graph and evaluate it on nodes E and D because we have data for these nodes.

- Inductive [71]: The input graph is divided into multiple subgraphs across splits (the dataset consists of multiple graphs that are independent of each other). The graph structure can only be observed by splitting it into its own parts, which allows the test to be performed and generalized to unseen graphs. By removing the dotted links in Figure 3 (f), we can see three distinct graphs: one for training, one for validation, and one for testing. As a result, when we make a prediction for a test set node F, we are no longer impacted by the structure information from other nodes. During training, we compute embeddings only using the nodes A, B, and C in the graph. During validation, we use the graph over nodes E and D to calculate the embeddings and evaluate the model on those nodes. However, the drawback of this approach is that splitting the original graph into many different small pieces results in the loss of some graph information.
- Scalability [72]: refers to the ability of models to efficiently and effectively handle the complexity of large-scale KGs, enabling efficient representation learning, semantic reasoning, and downstream applications such as link prediction, entity classification. The key aspects of scalability in KGE include:
  - Managing and processing a large amount of data that includes entities, relationships, and associated attributes.
  - Training on large datasets within a reasonable time frame, which involves enhancing training models, parallelizing computations, and utilizing distributed computing resources if necessary.
  - Efficiently handling large graphs by using sampling strategies such as random sampling, negative sampling, or neighborhood sampling techniques.
  - Employing model compression, quantization, or low-rank factorization to reduce the memory footprint of embedding models with-

- out sacrificing their representational power to enhance scalability.
- Managing computational resources such as the CPU and GPU is crucial for enhancing performance and scalability.

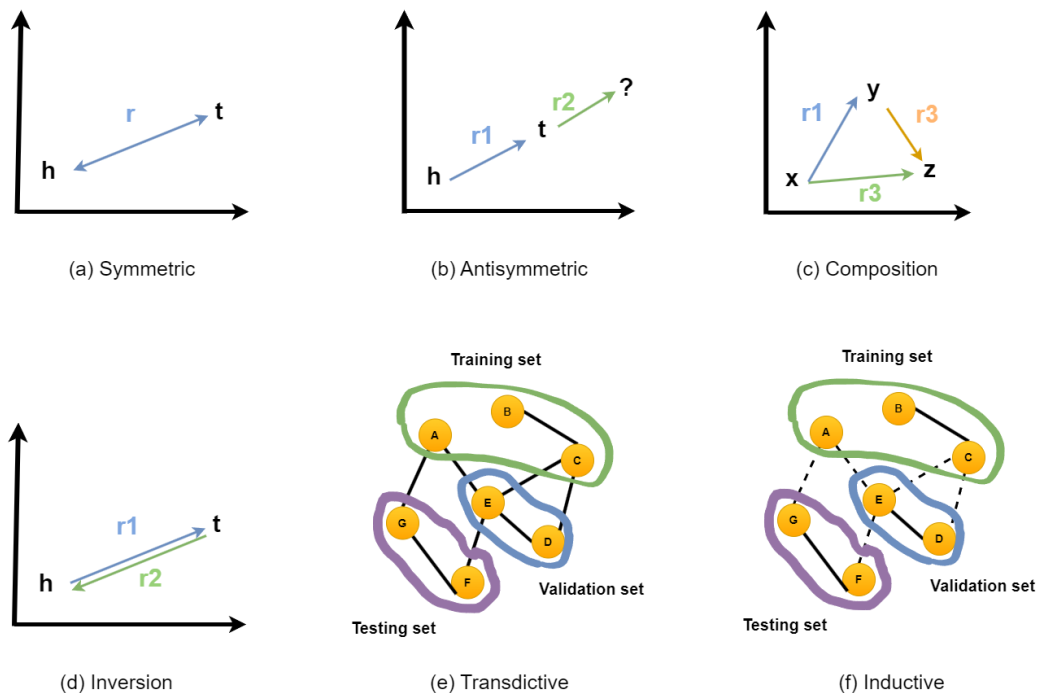


Figure 3: Presentation of different aspect of connection between entities

Developing a novel KGE model involves four essential steps. First, the representation space [69] must be defined, determining how entities and relations are represented. Second, a scoring function needs to be established to measure the plausibility of factual triples. Third, encoding models must be designed to represent and learn relational interactions effectively. Finally, auxiliary information, such as textual descriptions, relation/entity types, and entity images, should be incorporated into the proposed embedding methods to enhance the model’s accuracy and comprehensiveness.

The Euclidean point-wise space is the most widely used representation space [23], where entities are embedded in a vector space and interactions are modeled using vectors, matrices, or tensors (e.g., TransE, TransH, TransR, TransD, NTN, HolE, Analogy). Other representation spaces, such as complex

vector spaces, Gaussian distributions, and manifold spaces, are also explored [24].

In manifold space, point-wise embedding is relaxed, providing an advantage over Euclidean space. Gaussian embeddings can capture uncertainties in entities and relations, as well as multiple relation semantics (e.g., KG2E and TransG). Complex vector space embeddings are effective for modeling different relational connectivity patterns, especially symmetry/antisymmetry (e.g., ComplEx, RotatE).

Selecting the appropriate representation space is crucial for encoding the semantic information of entities and capturing their relational properties. When developing a representation learning model, it is important to choose and design the representation space carefully to match the nature of the encoding methods while balancing expressiveness and computational complexity.

Current research focuses on coding models, such as linear/bilinear models [70], factorization, and neural networks. Additionally, auxiliary information, including textual, visual, and typed information, plays a significant role.

Linear/bilinear models represent interactions between entities and relations using linear operations, such as SME, DistMult, ComplEx, and Analogy. Factorization methods formulate KGE models as three-way tensor decompositions, such as Rescal. Recent studies have shown that neural networks for encoding semantic matching have achieved remarkable predictive performance. Neural networks such as NTN, NAM, SME, ConvE, R-GCN, and KGAT offer alternative methods to encode models by incorporating linear/bilinear blocks.

## 4. Real world applications, datasets and evaluation metrics

### 4.1. Real-word applications

Knowledge Graphs (KGs) have received significant research attention in recent years. Depending on the focus, KG can be classified into two categories: KG construction techniques and KG applications.

KG construction technique research focuses on the representation, reasoning, and extraction of KGs, while application research aims to apply KGs to specific systems or domains. Based on the current survey, popular applications dedicated to applying KGs in specific areas include question answering systems (QA), recommender systems [82], and information retrieval systems. Additionally, KGs have wide applications in specific domains such

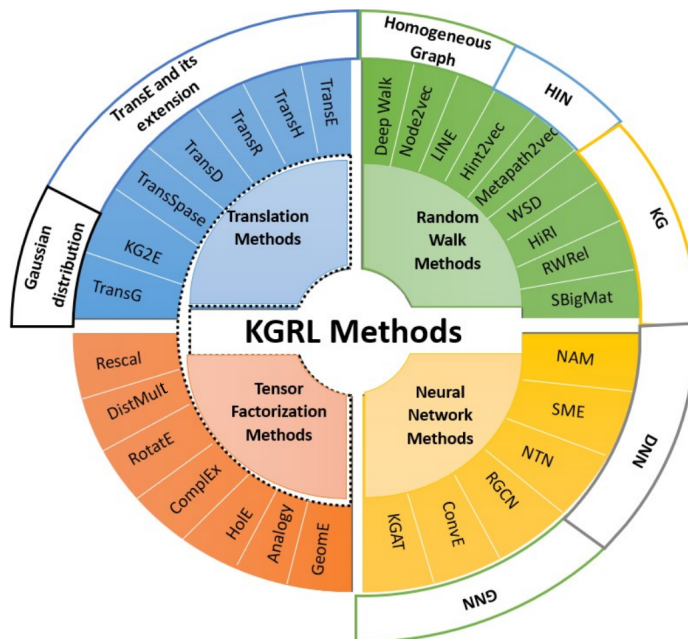


Figure 4: KGE taxonomy of methods

as medicine, finance, cybersecurity, news, and education. The taxonomy of KG application areas is shown in Figure 5.

#### 4.1.1. Question answering systems

In computer science, question answering is a traditional research topic that aims to find direct and precise answers from databases by analyzing users’ intentions through their queries. Today, natural language questions (NLQ) are answered with facts from KGs by the current KG-based question answering system [75]. In fact, the research results in semantic aware QA services are enhanced by the semantic information from KGs.

Among QA systems, Watson, developed by IBM to defeat human experts in the Jeopardy program, uses several knowledge bases, such as YAGO and DBpedia, as its data source. Social chatbots and digital assistants like XiaoIce, Cortana, and Siri also rely on structured knowledge as a key component. Freebase is the most well-known knowledge source used by many research QA systems, which often test their systems on WebQuestions.

Traditional QA systems over KGs [73] can be grouped into three categories: semantic parsing, information retrieval, and embedding. First ,

Table 3: Comparative table of KGE methods

Model	Sym	Antisym	Ind	Trans	Super	Semi	Unsup	Inv	Comp	Comlx
DeepWalk [40]	✓			✓	✓					$O(V \log V)$
Node2vec [41]		✓		✓	✓					$O(V \log V)$
LINE [43]	✓			✓			✓			$O(K * N)$
Hint2vec [44]				✓						$O(K * N)$
Metapath2vec [45]				✓			✓			$O(K * N)$
WSD [46]	✓			✓	✓					-
HiRi [49]				✓	✓					-
RWRel [47]				✓						-
SBigMat [48]	✓			✓	✓	✓	✓			-
TransE [28]		✓		✓	✓			✓	✓	$O(d)$
TransH [29]		✓	✓	✓	✓					$O(d)$
TransR [30]	✓	✓	✓	✓	✓			✓		$O(dk)$
TransD [31]	✓		✓	✓	✓			✓	✓	$O(\max(d, k))$
Transparse [53]	✓		✓	✓		✓		✓	✓	$O(dk)$
RotatE [58]	✓	✓	✓	✓	✓			✓	✓	$O(d)$
KGE [54]	✓			✓				✓		$O(d)$
TransG [56]	✓			✓	✓				✓	$O(dc)$
Rescale [57]	✓			✓	✓				✓	$O(d^2)$
DistMult [32]	✓		✓	✓						$O(d)$
ComplEx [34]	✓	✓	✓	✓	✓				✓	$O(d)$
HolE [33]	✓			✓	✓			✓		$O(d \log d)$
Analogy [35]	✓		✓	✓	✓		✓	✓		$O(d)$
GeomE [59]	✓			✓	✓			✓	✓	$O(d)$
NAM [62]		✓	✓							$O(Ld^2)$
SME [60]	✓		✓		✓					$O(d^2)$
NTN [61]		✓	✓							$O(d^2k)$
R-GCN [64]		✓	✓					✓		$O((n-1)K^2)$
ConvE [67]	✓		✓					✓		$O(d)$
KGAT [66]	✓		✓		✓					$O( G d^2)$

semantic parsing work transforms NLQs into logic forms that express the semantics of the entire query. The parsed results are then used to generate structured queries to search knowledge bases and obtain answers. Second, information retrieval consist to automatically translate NLQ into structured queries and retrieve a set of candidate answers from the kB. Features and

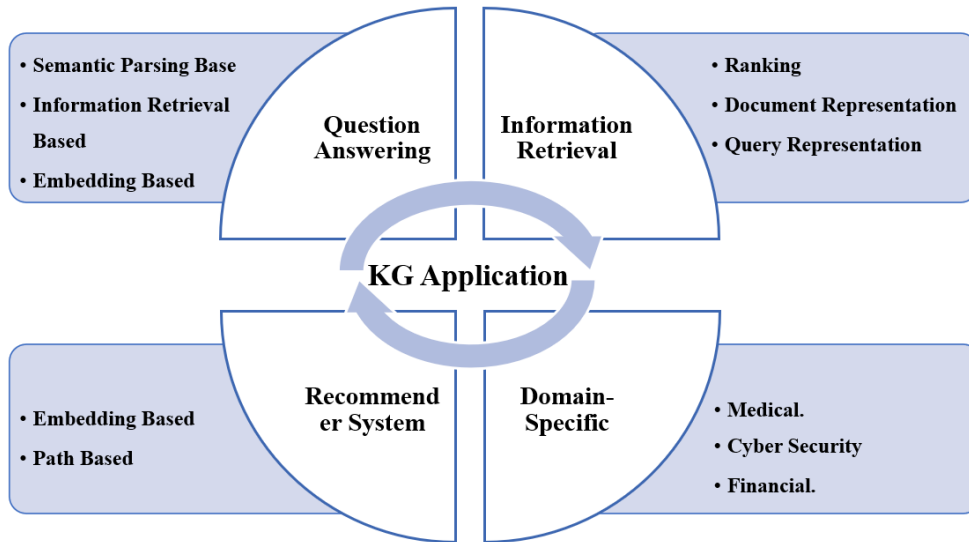


Figure 5: KG Applications taxonomy

candidates are extracted to rank these candidates, proposing the most likely correct answer. Finally, Embedding learns low-dimensional vector embeddings of given question and candidate answers. Then, the candidates with the highest similarity score will be considered as the final answers. Furthermore, in recent years, with the rapid growth of DL in the field of NLP, the performance of KG-based QA systems [76] has improved by combining deep learning methods with traditional methods.

#### 4.1.2. Recommender systems

The purpose of RSs [82] is to filter vital information fragments from a vast amount of dynamically generated data according to user preferences, interests, and behaviors. RSs gather information on a set of items (such as movies, songs, books, applications, and websites) and predict which item the user will prefer.

RSs have been successful implemented in many fields, including e-commerce websites and streaming services such as Netflix, as well as movie, music, and social networking sites. A wide variety of traditional approaches have been developed for building recommendation systems, including collaborative filtering, content-based filtering, and hybrid filtering. One of the most commonly used and mature filtering techniques is collaborative filtering [84].

Recently, studies have started to consider KGs as a source of side in-

formation. The relationships between different types in a KG help improve the precision of the recommender and increase the variety of recommended items [83]. The majority of methods used to construct KG-based recommender systems can be categorized into two approaches: embedding-based and path-based. The embedding-based approach preprocesses the KG using KGE methods and applies the learned entity embeddings to a recommendation framework. The path-based approach aims to create a graph model directly to explore various patterns of connections among nodes in the KG, providing additional information for recommendations.

#### *4.1.3. Information retrieval*

In a KG, retrieving relevant information is done by retrieving it from a structured and interconnected collection of data, where entities and their relationships are explicitly defined. KGs often contain a wealth of information, and effective information retrieval [85] is essential to extract valuable insights. The emergence of KGs has led to the incorporation of entity data from KGs by more and more commercial web-based search engines to improve their search results. For instance, Google uses data from Google Plus and Google KG, while Facebook uses Graph Search to search for entities. The property of KGs that contains human knowledge about real-world entities helps search systems improve their ability to understand queries and documents [86]. KGs' semantics can be utilized in various components, including query representation, document representation, and search system ranking.

#### *4.1.4. Domain-specific*

Among the new application domains in KGs, we discuss their use in the medical domain and the cybersecurity domain.

The development and application of KGs in the medical domain hold great promise in advancing healthcare [74], improving patient outcomes [77], and driving innovation in medical research and treatment. These KGs represent entities such as diseases, drugs, genes, and medical procedures, and their relationships in a structured format. The use of KGs in the medical domain offers numerous benefits, such as enabling advanced query capabilities, facilitating the integration of diverse data sources, and promoting personalized medicine. Applications of medical KGs include clinical decision support, drug discovery and development [89], disease profiling and stratification [88], patient cohort identification [87], and biomedical research.

The application of KGs in the cybersecurity domain [73] has gained significant attention due to the complex nature of cyber threats and the increasing need for efficient, accurate, and actionable intelligence. By using these KGs, security professionals can gain insights, detect patterns, and make informed decisions to improve overall security posture. A well-designed and maintained cybersecurity KG provides security teams with a holistic and contextualized view of their cybersecurity environment, empowering them to make informed decisions and respond effectively to cyber threats.

#### *4.2. Open access datasets*

The main goal in this section is to present available open access datasets for the scientific community. Table 4 presents a compilation of publicly accessible KGs, detailing their scale and composition. Notably, WordNet [90], established in the 1980s, serves as a comprehensive lexical database for English words. It features semantic connections such as synonymy, antonymy, hyponymy, meronymy, troponymy, and entailment, encompassing 155,000 entities and 207,000 facts. WordNet has two extensions: WN18 [91] and WN18RR [91].

In 2007, Freebase [92, 93] appeared as a large collaborative general knowledge base. It contains 44M entities and 2.4B facts. In Freebase, 71% of 3 million person entities miss a place of birth [104], 75% do not have a nationality, and 94% have no facts about their parents (West et al., 2014). FB15k and FB15k-237 are two subsets of Freebase. FB15k [94] is a large-scale KG containing general knowledge facts, mainly about movies, actors, awards, sports, and sports teams. FB15k-237 [94] is a subset of FB15k where the inverse relations are removed. Among these datasets, FB15k-237 and WN18RR are widely adopted for performance benchmarking, as they have removed the inverse relations to mitigate the issue of test leakage.

For example, both DBpedia [95] and YAGO [96, 97] are derived from information in Wikipedia infoboxes. In particular, English DBpedia 2014 contains 4.3M entities and 70M Triples, with 60% of person entities missing a place of birth and 58% of scientists lacking a fact about what they are known for. However, in 2008, YAGO also included information from GeoNames, which contains spatial information.

#### *4.3. Application tasks and evaluation metrics*

After introducing the existing KGE methods, real-world applications, and datasets, we explore various evaluation metrics that we classified according

Table 4: Open benchmarks datasets

Dataset	Description	Subset	#Ent	#Rel
WordNet (1980) [90]	Lexical database of semantic relations between words that contain 155 k of entities and 207 k of facts.	WN11	38,696	11
		WN18	40.943	18
		WN18RR	40.943	11
Freebase (2007) [92]	Large collaborative knowledge base for persons entities the birthplace that contains 44 M of entities and 2.4b of facts.	FB13K	75,043	13
		FB15K	14.951	1.345
		FB15K-237	14.951	237
DBpedia (2007) [95]	Automatically extracted structured data from Wikipedia that contain 3.4 M of entities and 70 M of facts	DB100K	99.604	470
YAGO (2008) [96]	An ontology and knowledge base that extends beyond Freebase includes information from GeoNames that contain 10 M of entities and 120 M of facts	YAGO3-10	123.182	37

to the application tasks. There are multiple types of application tasks employed to evaluate the performance of embedding models, with the most common methods including link prediction, Triple classification, and entity classification.

#### 4.3.1. Link Prediction

Link prediction [101] aims to predict missing relations and new links between entities in a KG given the existing links among entities. Several methods are used for link prediction in KGs, including translation-based methods (e.g., TransE, TransR, TransD, TransH), semantic matching-based methods (e.g., Rescal, ComplEx, DistMult), random walk-based methods, and neural network models (e.g., GNN and DNN). The performance of KGE methods can be quantified using standard evaluation metrics such as mean rank (MR), mean reciprocal rank (MRR), and hit-at-k.

- Mean rank (MR): A metric for evaluating the performance of a ranking algorithm or system, measuring the average position at which relevant

items appear in a ranked list.

- Mean Reciprocal Rank (MRR) [99] : Considers the rank of the first correct prediction. It is the reciprocal of the rank of the first correctly predicted positive example, giving higher scores for correct predictions that appear earlier in the ranked list.
- Area Under the Precision-Recall curve (AUC-PR) [100]: is applied as a performance metric to evaluate the effectiveness of binary classifiers, particularly in scenarios where class distributions are imbalanced. The AUC-PR provides a scalar value that summarizes the performance of a model by integrating the precision-recall curve, which plots precision against recall at various threshold settings.
- Hits@k [99] : The percentage of test triples where the ground truth entity ranks within the top K candidates.

#### 4.3.2. Triple classification

. Triple classification [103] in a KG involves predicting the correctness or validity of a triple (subject, predicate, object) within the graph. In a KG, triples represent relations between entities, and Triple classification aims to determine whether a given triple is true or false. Triples are labeled as positive (true) or negative (false) based on their correctness. Positive triples reflect real relationships in the underlying domain, while negative triples are incorrect or not present in the domain. Embedding models, such as TransE, TransH, or ComplEx, are used to learn vector representations of entities and relations. These embeddings capture semantic information about entities and relationships in the KG, and classification can be performed using machine learning models on top of these embeddings.

The evaluation metrics used in Triple classification are as follows:

- Precision at k (P@k) [102] : Measures the proportion of correctly predicted positive examples in the top k rankings. This metric is useful when only the top-ranked predictions matter.
- Recall at k (R@k) [102] : Measures the proportion of true positive examples in the top k predictions relative to all positive examples in the dataset. This metric is useful when identifying as many positive examples as possible is important.

- Accuracy [98] : Measured by the ratio of correctly predicted links to the total of non-existent links (only two-hop distances).
- Precision [102] : The percentage of true positive predictions out of all positive predictions. Precision values range from 0 to 1.
- Recall : Measures the proportion of total positives that were predicted to be positive.
- F1-Score [102] : Represents the harmonic mean between precision and recall. Both false positives and false negatives are taken into account in the F1 score.
- Area Under the Receiver Operating Characteristic curve (AUC-ROC) [98] : Measures the overall performance of a binary classification model. The AUC is calculated by adding the area under the ROC curve, and the larger the area, the more accurate the predictor. AUC-ROC is often used in binary classification problems.

#### 4.3.3. Entity classification

. Entity classification [103] involves assigning predefined classes or categories to entities based on their attributes, relations, or other relevant information within the graph. Entity classification uses supervised machine learning algorithms to learn a model that can predict entity classes based on input features. Common algorithms include decision trees, random forests, support vector machines, and neural networks. To evaluate the performance of this task, we use the same metrics as in Triple classification, which are precision, accuracy, recall, F1-score, and ROC-AUC.

#### 4.4. Source codes and libraries

There are several libraries and frameworks available for processing KG and KNN, each offering unique features and capabilities. AmpliGraph focuses on KGE, providing tools for link prediction, node classification, and clustering. One popular library is NetworkX, a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. For more advanced tasks such as embedding and link prediction, PyTorch Geometric (PyG) which is a scalable GNN library, that offer a variety of graph operations and utilities and DGL (Deep Graph Library) supports multiple backends (PyTorch, MXNet, TensorFlow) and is

designed for scalable GNNs with an extensive API. These libraries provide efficient and scalable implementations of various graph neural network (GNN) models and can leverage GPU acceleration. StellarGraph is another useful library specifically designed for ML on graph-structured data, providing implementations of popular algorithms like GraphSAGE, GCN, and HinSAGE. PyKEEN (Python KnowEdge EmbeddiNGs) is a Python library designed to train and evaluate KGE models that can incorporate multi-modal information. Graph-tool is a Python library that offers efficient algorithms for graph analysis and manipulation, known for its performance and ability to handle large-scale graphs. DGL-KE is a high performance, easy-to-use, and scalable package for learning large-scale KGE. The package is implemented on the top of DGL and developers can run DGL-KE on CPU machine, GPU machine, as well as clusters with a set of popular models. Collectively, these libraries provide a robust toolkit for researchers and practitioners working with KGs, enabling efficient processing, analysis, and application of graph-based data. Table 5 provides a summary of the creation dates, creators, build type and the link of the specified libraries.

Table 5: Open-source libraries

Library	Creation	Creator	Build	Link
NetworkX	2002	Aric Hagberg, Dan Schult, and Pieter Swart	Python	<a href="https://networkx.github.io/">https://networkx.github.io/</a>
Graph-tool	2006	Tiago de Paula Peixoto	C++ with a Python	<a href="https://graph-tool.skewed.de/">https://graph-tool.skewed.de/</a>
AmpliGraph	2018	Accenture Labs	TensorFlow	<a href="https://github.com/Accenture/AmpliGraph">https://github.com/Accenture/AmpliGraph</a>
PyG	2018	TU Dortmund University	PyTorch	<a href="https://github.com/pyg-team/pytorch_geometric">https://github.com/pyg-team/pytorch_geometric</a>
DGL	2019	NYU, AWS, and others	PyTorch/ MXNet/ TensorFlow	<a href="https://www.dgl.ai/">https://www.dgl.ai/</a>
StellarGraph	2019	CSIRO’s Data61	TensorFlow / Keras	<a href="https://stellargraph.readthedocs.io/">https://stellargraph.readthedocs.io/</a>
PyKEEN	2020	University of Bonn and Fraunhofer IAIS	PyTorch	<a href="https://github.com/pykeen/pykeen">https://github.com/pykeen/pykeen</a>
DGL-KE	2020	AWS AI	PyTorch/ MXNet/ TensorFlow	<a href="https://github.com/awsml/dgl-ke">https://github.com/awsml/dgl-ke</a>

## 5. Experimentation

### 5.1. Methodology

In this section, we provide a comprehensive report on the experiments and comparisons conducted in our work. Our experiment focuses on link prediction as a downstream task. The main objectives of this experimental methodology are to investigate whether different KGE models can predict missing links within a KG effectively, efficiently, and inductively, and to assess their effectiveness across different metrics. We aim to explain the comparative analysis of selected KGE models, detailing their underlying scoring functions, training procedures, and the specific datasets used in the experiments. Understanding the strengths and weaknesses of various KGE models is crucial for advancing KG research. By identifying the best models for different scenarios, researchers and practitioners can make informed decisions about which models to use in experimentation, thereby enhancing the accuracy and efficiency of KG applications.

#### 5.1.1. Experimental set-up

Our analysis begins with a concise overview of the environment we use for our analysis, as well as the methods we employ to train and evaluate the selected KGE models.

Our experiments, training, and evaluation of each model are conducted on the Grid5000 <sup>1</sup> server environment using a cluster with 3 nodes, 6 AMD EPYC 7352 (Zen 2) CPUs, x86\_64 architecture, 2 CPUs per node, 24 cores per CPU, 256 GB RAM, and 2 Nvidia A40 GPUs (45 GiB). The operating system is Ubuntu 20.04. The implementation of the model architectures is based on the PyTorch framework.

We train and evaluate some selected models introduced in Section 2. From the family of translation methods, we chose TransE and RotatE. From the family of tensor factorization methods, we selected DistMult and ComplEx. From the last family, we selected the R-GCN model. To ensure our results are reproducible, we use publicly available DGL implementations <sup>2</sup> whenever possible.

We adopt four widely used KG datasets to evaluate performance, which can be classified into four categories: FB15k, FB15-237, WN18, and WN18RR.

---

<sup>1</sup><https://www.grid5000.fr/w/Grid5000:Home>

<sup>2</sup><https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding>

These datasets allow for a comprehensive evaluation of the models’ performance. The details of these four datasets are shown in Section 3.1. Table 6 summarizes their full statistical information on data splitting, including training, test, and validation.

Table 6: Data Description

Dataset	#Train	#Test	#Val
FB15K	483142	59071	50000
WN18	141442	5000	5000
FB15K-237	272115	20466	17535
WN18RR	86835	3134	3034

To assess the performance of the indicated methods, we report global results for each model using the five most popular metrics for link prediction, which we present in Section 3.3: H@1, H@3, H@10 and MRR

### 5.1.2. Baseline analyses

In this section, we conduct a set of experiments. First, we present our results on the predictive performance of KGE models in link prediction task. The goal of these experiments is to analyze how predictive performance varies when considering the type of relations: symmetric/anti-symmetric, inversion, composition, and whether the model can handle one-to-many (1-n) or many-to-many (n-n) relations. Additionally, we consider the size of the dataset (number of entities and relations).

In the second part, we evaluate the efficiency of KGE models in link prediction task in terms of training and prediction time, and assess the scalability of the best models in each category. We evaluate each model’s efficiency based on two primary factors:

- Training time: the time needed to train and optimize embeddings for relationships and entities.
- Prediction time: the time needed to produce comprehensive rankings for a single test fact, encompassing both head and tail predictions.

In the third part, we discuss the inductivity of the tested models and we present the categories that has the best inductivity performance. The main focus of inductive representation is on how well the model represents unseen

entities. The evaluation ability of the model increases with the number of unseen entities in the evaluation dataset. As a result, the experiment uses the GraiL dataset division method<sup>3</sup> to divide the dataset. Root nodes are randomly chosen from the KG, and the training graph is composed of K-hop neighborhood nodes around the root node. Following the removal of the training graph from the KG, the test graph is generated similarly, where each dataset is divided into four parts: 1, 2, 3, and 4. The corresponding test sets are ind1, ind2, ind3, and ind4. The model’s inductive characteristics can be verified by ensuring that there are no duplicate entities in both training and test sets, and by comparing the relations in the test graph to those in the training graph.

Finally, we present the CPU, GPU and RAM energy and power consumption, then total electricity consumption, and CO2 emissions that we described in details:

- CPU Energy and Power Consumption <sup>4</sup> [78]: the energy consumption of a CPU varies based on its usage, clock speed, and architecture. Modern CPUs can consume anywhere from a few watts (idle) to over 200 watts (under heavy load).
- GPU Energy and Power Consumption <sup>5</sup> [78]: GPUs typically consume more power than CPUs due to their parallel processing capabilities. Power consumption ranges from tens of watts for low-end models to over 300 watts for high-end models.
- RAM Energy and Power Consumption [79]: RAM power consumption varies with the type (DDR3, DDR4, etc.), size, and number of modules. It generally ranges from 1.5 to 5 watts per module.
- Total Electricity Consumption [80]: total electricity consumption of a computer system is the sum of the power consumed by the CPU, GPU, RAM, and other components (motherboard, storage, peripherals).
- CO2 Emissions [81]: CO2 emissions are calculated based on the electricity consumption and the carbon intensity of the electricity source.

---

<sup>3</sup><https://github.com/kkteru/grail/tree/master/data>

<sup>4</sup><https://gamernexus.net/cpus/intel-problem-cpu-efficiency-power-consumption>

<sup>5</sup><https://www.nvidia.com/en-us/glossary/power-efficiency/>

## 5.2. Experimental Results

### 5.2.1. Effectiveness

In this section, we report the results regarding the effectiveness of KGE models in the link prediction task in terms of time and predictive performance. Our objective for this experiment is to examine the variations in predictive performance when considering the size of the dataset, the complexity, and the type of relations. We present our results in Figures 6 (a), 6 (b), 6 (c), and 6 (d). We use MRR, Hits@1, Hits@3, and Hits@10 for measuring effectiveness.

Table 7: The performance results of KGE models

Dataset	FB15K				WN18			
Metric	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	0.2759	0.1593	0.3127	0.5277	0.418	0.1805	0.602	0.814
RotatE	<b>0.2798</b>	<b>0.1642</b>	<b>0.3166</b>	<b>0.5285</b>	<b>0.6043</b>	<b>0.483</b>	<b>0.6826</b>	<b>0.8311</b>
CompLex	0.2252	0.1168	0.2557	0.4568	0.030	0.011	0.013	0.050
DistMult	0.0983	0.0399	0.1004	0.2112	0.032	0.013	0.012	0.005
R-GCN	0.1002	0.0666	0.0958	0.1571	0.0469	0.0247	0.0441	0.0851
Dataset	FB15K-237				WN18RR			
TransE	0.1698	0.1001	0.1730	0.3181	0.1429	0.0015	0.2268	0.4430
RotatE	<b>0.1766</b>	<b>0.1061</b>	<b>0.1820</b>	<b>0.3255</b>	<b>0.3059</b>	<b>0.01940</b>	<b>0.3814</b>	<b>0.5036</b>
CompLex	0.0814	0.0281	0.0820	0.1881	0.060	0.010	0.019	0.006
DistMult	0.0216	0.0098	0.0265	0.0401	0.061	0.010	0.020	0.062
R-GCN	0.1064	0.0686	0.1049	0.1715	0.0405	0.0202	0.0352	0.0789

Based on Figure 6 and Table 7, RotatE achieves the best performance in all benchmark datasets. This results are due to that the RotatE can capture the different type of relations like the inversion, the composition, the symmetry and anti-symmetry relations that we indicate in Table 3. This model can, also, easily calculate and generalize a large different type of relation which we have FB15K has 1345 types of relations compared to other datasets which have few relation types that we present in Table 4. In contrast, R-GCN, on the other hand, ranks third with WN18, FB15K-237 and WN18RR, but not with FB15K. This is because R-GCN must gather all information about neighbors that have one or more relations. However, the performance of

FB15K has declined due to the exponential increase in computational complexity for R-GCN, which has 1345 types of relations. The second best rank is for TransE, we can explain that with TransE can't resolve the symmetric relation (Table 3) and the as trouble coping with complex relations that an entity can have multi-relations (Section 2.2.1). To better compare the performance of different models on each evaluation metrics, Figure 6 displays the curves drawn for every model in relation to each benchmarks with diverse metrics.

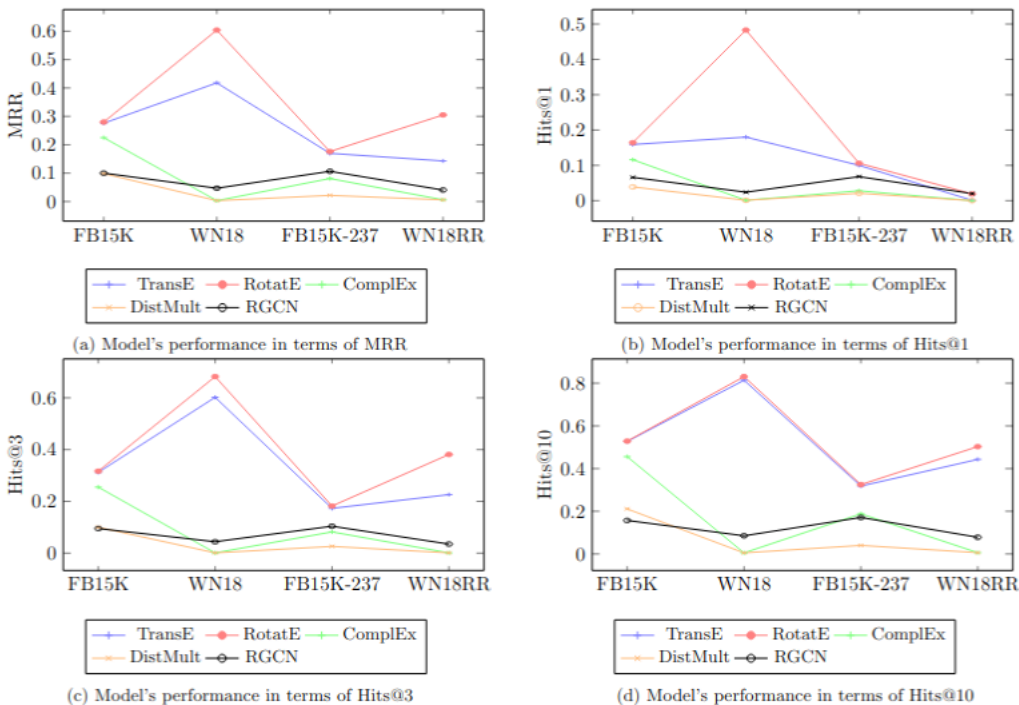


Figure 6: Performance variation of different models.

For the other models, ComplEx and DistMult exhibit minimal performance. This can be attributed to the fact that DistMult can only capture symmetric relations, whereas ComplEx is more effective than DistMult as it can capture both composed and symmetric/anti-symmetric relations (details in Table 3 in Section 2.4).

We can also observe that the performance of the models decreases with the datasets FB15K-237 and WN18RR compared to FB15K and WN18. This can be explained by the size of the data and the types of relations

having a significant impact on performance. Additionally, this result can be attributed to the removal of inverse relations, which leads to the loss of important information. Therefore, based on these results, we can conclude that the literature and experimental analyses are well aligned.

### 5.2.2. Efficiency

In this section, we present our findings on the efficiency of KGE models in the link prediction task in terms of training and prediction time. In Figure 7 and Table 8, we illustrate the training time in seconds for each model on each dataset. We observed that training times vary from a few seconds for R-GCN to about 3 hours for other models. Compared to the embedding-based models, the R-GCN model is strikingly fast. R-GCN considers the training time as a configuration parameter and achieves optimal performance within 14 to 15s for FB15k and WN18, and 16 seconds for FB15k-237 and WN18RR. To explain this results, the R-GCN model requires a high amount of memory, which is preserved by the Grid5000 server’s high performance, which enables faster training and prediction time.

In Figure 8 and Table 8, we depict the prediction time for each model, which refers to the duration needed to generate complete rankings for both h and t predictions of a single fact. These scores are primarily influenced by the dimensions of the embeddings and the size of the evaluation batch. In our experiments with these models, we used a maximum evaluation batch size of 1024, which is the maximum allowed in the setting. We observed that the R-GCN model has the best training and prediction times with all benchmark datasets compared to the other models, where ComplEx ranks second and DistMult ranks third.

Training and prediction times are influenced by several factors, including the model architecture, with deep neural networks such as R-GCN typically requiring longer computations. Additionally, model hyperparameters, such as embedding size and the number of negative samples per positive instance, play a crucial role. The size of the dataset, encompassing the number of entities, relations, and training triples to process, also significantly impacts these times. For example, Freebase has a wide variety of relations and a large volume of facts and entities compared to other datasets, resulting in longer training and prediction times. Additionally, the type of relations influences the time required; datasets like FB15K-237 and WN18RR, which have removed inverse relations, experience faster training and prediction. We further investigate how the training times of the best-performing models

within each family scale as the volume of data increases.

Table 8: Training and prediction time results

Dataset	FB15K		WN18		FB15K-237		WN18RR	
	Train	Pred	Train	Pred	Train	Pred	Train	Pred
TransE	9393.68	62.9726	2773	18.55	5384.39	35.18	1698.38	11.27
RotatE	7708.13	51.19	2598.76	17.31	4309.52	28.69	1593.61	10.62
CompLex	6177.28	41.13	1704.46	11.21	3944.04	26.04	1051.78	7.01
DistMult	7141.95	47.63	2056.67	13.66	3983.088	28.11	1230.62	8.099
R-GCN	14.49	0.065	15.24	0.0956	16.39	0.0924	16.18	0.10

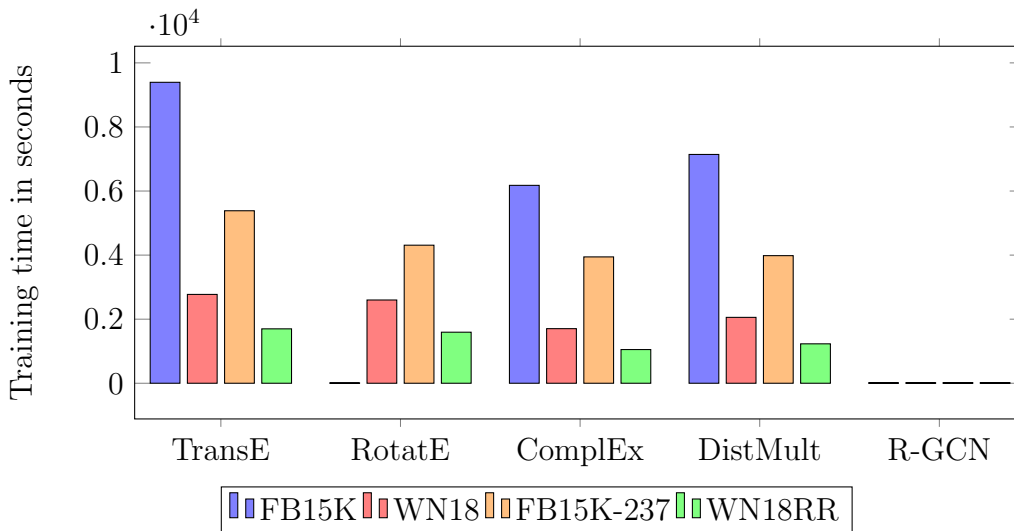


Figure 7: Training time

### 5.2.3. Inductivity

Our goal in this section is to test and evaluate the tested model by completing link prediction tasks and adopting Hits@10 as the evaluation metric to evaluate the inductivity performance of each models. To run experiments on the GraIL method, we used the code <sup>6</sup> described in the ICML’20 paper [105].

<sup>6</sup><https://github.com/kkteru/grail>

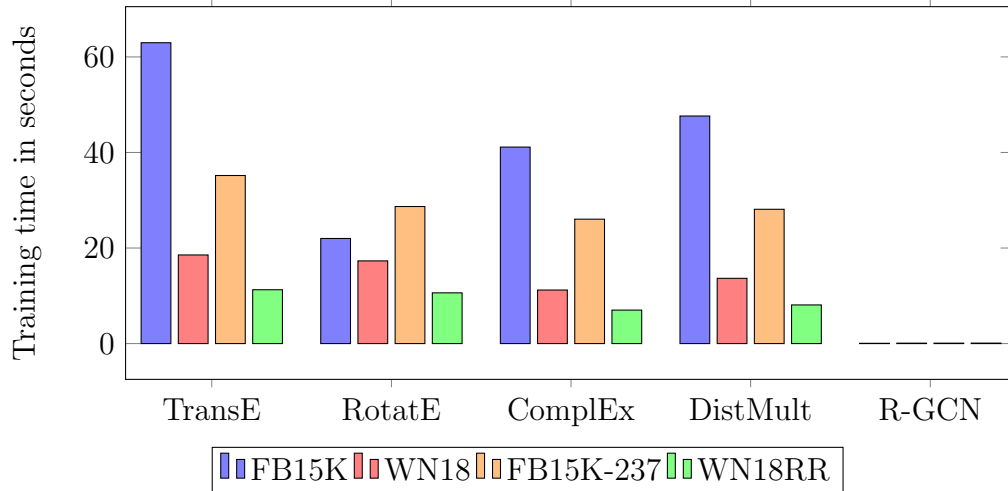


Figure 8: Prediction time

Table 9 demonstrates the experimental results of Hits@10 on datasets 1, 2, 3, and 4 of WN18RR and FB15K-237. According to the results presented in Table 9 and Figure 9, we observe that R-GCN achieves the best results with the two benchmarks across all four test sets. This can be explained by the fact that the R-GCN model is naturally inductive. For the other models, we observe that RotatE, ComplEx, and DistMult have acceptable results due to their ability to capture complex interactions between entities and relations and their capacity to model intricate relationships. TransE shows lower inductivity compared to models like ComplEx or RotatE due to its simplistic modeling of relations, which may limit its ability to handle nuanced reasoning tasks.

Table 9: Inductivity results

Dataset	FB15K-237					WN18RR				
	TransE	RotatE	Complex	DistMult	R-GCN	TransE	RotatE	Complex	DistMult	R-GCN
1	0.09	0.35	0.23	0.20	0.59	0.05	0.23	0.21	0.18	0.80
2	0.082	0.27	0.18	0.19	0.55	0.04	0.21	0.19	0.17	0.76
3	0.03	0.21	0.15	0.15	0.52	0.01	0.17	0.17	0.14	0.70
4	0.01	0.17	0.145	0.10	0.45	0.01	0.16	0.14	0.11	0.48

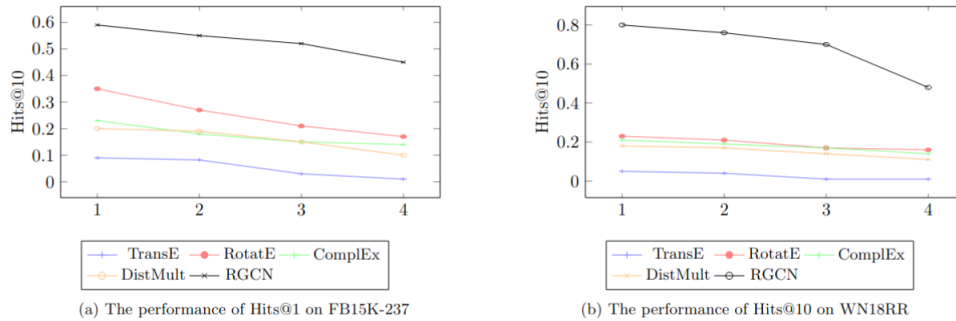


Figure 9: Inductivity results

#### 5.2.4. CO2 Emissions

Green computing plays a critical role in reducing the environmental impact of information technology (IT) and computing. The aim of this field is to design, manufacture, use and dispose of computers, servers, and associated subsystems efficiently and effectively without causing any environmental impact. IT and computers, including data centers, communication networks, and user devices, account for an estimated 4-6% [107] of global electricity consumption. Data centers alone consume about 1-1.5% [107] of global electricity. Although the energy consumption of IT and computers is substantial but not as high as 11% [108] compared to airplanes and cars, it contributes significantly to climate change due to the CO2 emissions generated by energy consumption. In terms of CO2 emissions, the ICT sector contributes around 2-3% of global greenhouse gas (GHG) emissions [109]. The energy required to train complex models in data centers leads to substantial carbon footprints, exacerbating pollution, and global warming.

In this section, we tested the effect of the tested models in terms of power and energy consumption, electricity, and CO2 emissions. Based on Figure 10, the analysis reveals that DistMult and R-GCN consistently demonstrate superior efficiency across all metrics, including RAM power consumption, GPU energy and power consumption, CPU power and energy consumption, total electricity consumption, and CO2 emissions. TransE, on the other hand, exhibits the highest consumption and emissions in these categories. The characteristics of each model can explain these results. TransE, being a translational distance model, requires more computational resources due to its simplicity and the need for extensive parameter tuning to achieve competitive performance. RotatE and ComplEx, which are more complex models

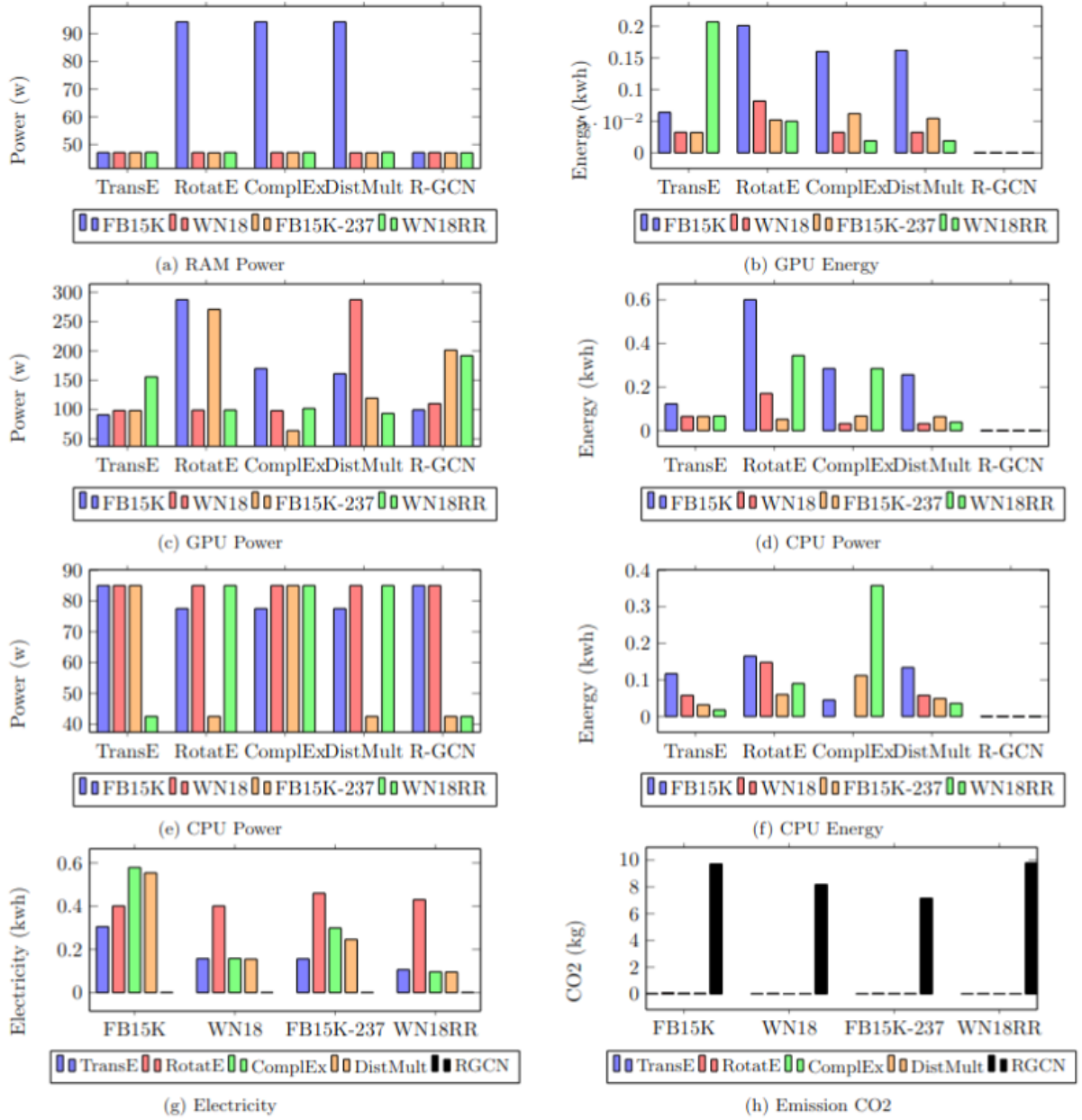


Figure 10: CPU, GPU, RAM, electricity estimation and emission of CO2

involving rotation and complex number embeddings, respectively, offer moderate efficiency. DistMult, a bilinear model, is computationally efficient due to its simplicity and effective handling of multirelational data without extensive parameterization. R-GCN, a graph convolutional network model, leverages graph structures to efficiently process relational data, resulting in lower power and energy consumption and reduced CO2 emissions. Consequently, the inherent simplicity and structural advantages of DistMult and R-GCN make them the most efficient models in terms of computational and environmental metrics.

## 6. Future directions

There are several interesting research directions to explore in future studies related to KG. In this section, we highlight some future directions that is based on the literature.

### 6.1. *Explainable recommendation for knowledge graph*

Despite the use of unstructured data, explainable recommendations [110] are mostly based on text, images, audio, video stills, and so on. However, if the RS possesses knowledge about the recommendation domain, it can generate personalized recommendations and explanations.

The advancement of KGE has enabled the integration of graph embedding learning with recommendation techniques, enhancing the explainability of recommendations. Consequently, the system can recommend items based on domain knowledge and provide users with understandable explanations for their recommendations. For example, in [111], a recommendation method was developed that predicts interactions between users and items using a KG and review text. In this instance, explanations are generated based on the paths between a user and an item. Similarly, in [112], a KG is used to explain recommendations to users through unstructured textual descriptions of items.

Future research in intelligent systems should focus on developing KG-based models for explainable recommendations. These models can provide personalized recommendations in various fields, including medical care, online education, and conversational systems, thereby increasing trust, transparency, and user acceptance in KG-based recommendation system responses.

### 6.2. Multi-task learning

G-based RS can be understood as link prediction in graphs. By considering the nature of the knowledge graph (KG), it is possible to enhance the performance of graph-based recommendations. For instance, missing facts in the KG can lead to missing relations or entities. If these facts are missing, the user’s preferences may be overlooked, negatively impacting the recommendation results. Studies such as [113], [114] have demonstrated that training both the KGC module and the recommendation module jointly can result in better recommendations. Other experiments have employed multi-task learning [72] by involving the recommendation module in the KGE task and the item relation regulation task [115], [116].

### 6.3. Large language models

Large Language Models (LLMs) [117], like GPT variants, present significant breakthroughs in Natural Language Processing (NLP) due to their substantial capacity for creation and generalization. Despite this, they are often considered black boxes, containing implicit knowledge within their parameters and exhibiting limitations in factual understanding and interpretation. On the other hand, KGs, as traditional knowledge models, offer explicit data storage but face shortcomings in construction, implicit relationship inference, and dynamic programming for incomplete data.

By combining their strengths[118], LLMs and KGs can enhance each other in AI applications. While KGs are efficient in managing and retrieving structured information, LLMs are highly skilled in comprehending and creating natural language, managing unstructured data. Integrating KGs allows LLMs to have access to accurate and structured data, which can enhance contextual understanding and relevance for tasks such as QE. KG can be built and enriched by LLMs by extracting information from large text corpora [119]. Together, they allow for more robust AI systems that can understand complex language and reason logically.

### 6.4. Dynamic knowledge graphs

Traditional KGs are often static and updated periodically. In the future, KGs will become dynamic, evolving continuously to incorporate real-time data and changes. By including temporal aspects, Dynamic Knowledge Graphs (DKGs) [120] extend traditional static KGs. They gather information about entities, relations, and events that change over time. Characteristics of

DKGs include temporal annotations on graph elements, versioning or History tracking, and support for querying and reasoning with temporal constraints.

However, Dynamic Knowledge bases face challenges related to the scalability and efficiency of temporal query processing, the management of Historical data through versioning strategies, the maintenance of consistency in distributed or federated databases, and the handling of uncertainty in temporal knowledge. In this sense, research efforts are focused on developing efficient temporal reasoning algorithms, distributed temporal graph databases, and incorporating machine learning techniques for temporal pattern recognition in DKGs.

### *6.5. Multi-modal knowledge graphs*

Multi-modal KG (MMKG) [121] integrate various types of data, such as text, images, audio, and video, to provide a more comprehensive and enriched representation of knowledge. Unlike traditional KG, which typically rely solely on textual information, MMKG incorporate diverse data sources to capture the complexity and richness of real-world information. This integration enhances the ability to perform tasks like entity recognition, relation extraction, and RS by leveraging the complementary nature of different data modalities. The multi-modal [122] approach allows for a deeper understanding of entities and their relations, facilitating more accurate and context-aware applications. For instance, by combining textual descriptions with visual data, a MMKG can improve image captioning and visual question answering tasks. Such advancements open new avenues for research and practical applications, making MMKG a promising area of study in AI and data science.

## **7. Conclusion**

This paper has provided a comprehensive overview of KGE methods. Through detailed comparative analysis, we have summarized the existing mainstream KGE, highlighting their strengths and weaknesses as discussed in the literature. We classified these methods into four families: random walk-based methods, translation-based methods, tensor factorization-based methods, and deep neural network-based methods.

Our experimental evaluations demonstrated the performance of selected KGE methods across various use cases, focusing on their effectiveness, efficiency, and inductive performance. Furthermore, the experiments under-

scored the importance of the characteristics of the dataset and the hyperparameters of the model in influencing the effectiveness of KGE methods and training and prediction times.

Looking ahead, we have identified several promising research directions in the field of KGE. These include the development of dynamic Kgs, the integration of large language models with KGs, and the enhancement of explainability in recommendation systems using KGs. Addressing the challenges associated with scalability, data sparsity, incompleteness, and missing links will be crucial to advancing the capabilities of KGE.

## References

- [1] Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* 2017, 29, 2724–2743.
- [2] R. Logan, N. F. Liu, M. E. Peters, M. Gardner, and S. Singh, “Barack’s wife hillary: Using knowledge graphs for fact-aware language modeling,” in *ACL*, 2019, pp. 5962–5971
- [3] Wang, R.; Wang, M.; Liu, J.; Chen, W.; Cochez, M.; Decker, S. Leveraging Knowledge Graph Embeddings for Natural Language Question Answering. In *Proceedings of the DASFAA 2019*, Chiang Mai, Thailand, 22–25 April 2019; pp. 659–675.
- [4] Musto, C.; Basile, P.; Semeraro, G. Embedding Knowledge Graphs for Semantics-aware Recommendations based on DBpedia. In *Proceedings of the UMAP 2019*, Larnaca, Cyprus, 9–12 June 2019; pp. 27–31.
- [5] ZHE CHEN , YUEHAN WANG, BIN ZHAO, JING CHENG, XIN ZHAO, AND ZONGTAO DUAN. Knowledge Graph Completion: A Review. *IEEE Access*, pp. 192435 - 192456, 2020.
- [6] ] J. Li and L. Hou, ”Review of knowledge graph research,” *J. Shanxi Univ., Natural Sci. Ed.*, vol. 40, no. 3, pp. 454–459, Mar. 2017.
- [7] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, ”Freebase: A collaboratively created graph database for structuring human knowledge,” in *Proc. ACM SIGMOD Int. Conf. Manage. Data SIGMOD*, 2008, pp. 1247–1250.
- [8] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, ”DBpedia: A nucleus for a Web of open data. ” in *The Semantic Web*. Berlin, Germany: Springer, 2007, pp. 722–735.
- [9] F. M. Suchanek, G. Kasneci, and G. Weikum, Yago: A core of semantic knowledge,” in *Proc. 16th Int. Conf. World Wide Web WWW*, 2007, pp. 697–706.
- [10] J. Ding and W. Jia. The research advances of knowledge graph completion algorithm. *Inf. Commun. Technol.*, no. 1, pp. 56–62, Apr. 2018.

- [11] S. C. Shapiro. A net structure for semantic information storage, deduction, and retrieval. in Proc. 2nd Int. Joint Conf. Artif. Intell., Los Altos, CA, USA, Sep. 1971, pp. 512–523.
- [12] Zhiqiang Zhong, Anastasia Barkova, Davide Mottin. Knowledge-augmented Graph Machine Learning for Drug Discovery: A Survey from Precision to Interpretability, arXiv:2302.08261v2 [cs.LG] 7 Mar 2023.
- [13] Tian, Ling, et al. "Knowledge graph and knowledge reasoning: A systematic review." *Journal of Electronic Science and Technology* 20.2 (2022): 100159.
- [14] Shen, Tong, Fu Zhang, and Jingwei Cheng. "A comprehensive overview of knowledge graph completion." *Knowledge-Based Systems* 255 (2022): 109597.
- [15] Martinez-Gil, J. et al.. Examining the Adoption of Knowledge Graphs in the Manufacturing Industry: A Comprehensive Review. In: Soldatos, J. (eds) *Artificial Intelligence in Manufacturing*. Springer, Cham. 2024.
- [16] Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, Shangsong Liang. Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces, arXiv:2211.03536, 2023.
- [17] Bram Steenwinckel, Gilles Vandewiele, Michael Weyns, Terencio Agozzino, Filip De Turck, and Femke Ongenaes. Ink: knowledge graph embeddings for node classification. *Data Mining and Knowledge Discovery*, 36(2):620–667, 2022.
- [18] Shreyansh Bhatt, Swati Padhee, Amit Sheth, Keke Chen, Valerie Shalin, Derek Doran, and Brandon Minnery. Knowledge graph enhanced community detection and characterization. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 51–59, 2019.
- [19] Meihong Wang , Linling Qiu and Xiaoli Wang. A Survey on Knowledge Graph Embeddings for Link Prediction, *Symmetry* 2021, 13, 485
- [20] Vignesh Prabhakara, Chau Vu, Jennifer Crawford, Joseph Waite, Kai Liue. An Ensemble Learning Approach to perform Link Prediction on

Large Scale Biomedical Knowledge Graphs for Drug Repurposing and Discovery, bioRxiv, 2023

- [21] Matteo Palmonari and Pasquale Minervini. Knowledge graph embeddings and explainable ai. Knowledge Graphs for Explainable Artificial Intelligence: Foundations, Applications and Challenges, 47:49, 2020.
- [22] Md Kamrul Islam , Diego Amaya-Ramirez , Bernard Maigret , Marie-Dominique Devignes , Sabeur Aridhi and Malika Smaïl-Tabbone. Molecular-evaluated and explainable drug repurposing for COVID-19 using ensemble knowledge graph embedding, Scientific Reports, (2023) 13:3643.
- [23] Yuanfei Dai, Shiping Wang, Neal N. Xiong, Wenzhong Guo. A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks. Electronics 2020, 9, 750, 2 May 2020.
- [24] Quan Wang, Zhendong Mao, Bin Wang, Li Guo. Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Transactions on Knowledge and Data Engineering. olume: 29, Issue: 12, 2724 - 2743, 01 December 2017.
- [25] Shaoxiong Ji, Shirui Pan, Member, Cambria, Senior, Pekka Marttinen, Philip S. Yu. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. IEEE Transactions on Neural Networks and Learning Systems, Volume: 33, Issue: 2, p 494 - 514, February 2021.0
- [26] Xiou Ge, Yun-Cheng Wang, Bin Wang and C.-C. Jay Kuo. Knowledge Graph Embedding : An Overview. arXiv:2309.12501v1 [cs.AI] 21 Sep 2023
- [27] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Martinata, and Paolo Merialdo. Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. ACM Transactions on Knowledge Discovery from Data Vol. 15, No. 2, 2021.
- [28] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. Advances in Neural Information Processing Systems 26 (NIPS), 2013.

- [29] Zhen Wang<sup>1</sup>, Jianwen Zhang, Jianlin Feng, Zheng Chen. Knowledge Graph Embedding by Translating on Hyperplanes. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, June 2014.
- [30] Yankai Lin, Zhiyuan Liu<sup>1</sup>, Maosong Sun, Yang Liu, Xuan Zhu. Learning Entity and Relation Embeddings for Knowledge Graph Completion. Twenty-Ninth AAAI Conference on Artificial Intelligence, Vol. 29 No. 1, 2015.
- [31] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, Jun Zhao. Knowledge Graph Embedding via Dynamic Mapping Matrix. 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), p 687–696, July, 2015.
- [32] Yang, B.; Yih, S.W.t.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. In Proceedings of the 2015 International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
- [33] Nickel, M.; Rosasco, L.; Poggio, T. Holographic Embeddings of Knowledge Graphs. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1955–1961.
- [34] rouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2071–2080.
- [35] H. Liu, Y. Wu, and Y. Yang, “Analogical inference for multirelational embeddings,” in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 2168–2178.
- [36] Feng Xia, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan and Xiangjie Kong (9 Aug 2020). Random Walks: A Review of Algorithms and Applications. arXiv:2008.03639v1 [cs.SI].
- [37] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications, IEEE transactions on knowledge and data engineering, VOL. XX, NO. XX, SEPT 2017.

- [38] William L. Hamilton. Graph Representation Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Vol. 14, No. 3 , Pages 1-159, 2020.
- [39] William L. Hamilton, Rex Ying, Jure Leskovec. Representation Learning on Graphs: Methods and Applications, IEEE Computer Society Technical Committee on Data Engineering, 2017.
- [40] Bryan Perozzi, Rami Al-Rfou, Steven Skiena. DeepWalk: Online Learning of Social Representations. KDD '14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining August 2014
- [41] Aditya Grover, Jure Leskovec. node2vec: Scalable Feature Learning for Networks. arXiv:1607.00653v1 [cs.SI] 3 Jul 2016.
- [42] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining August 2016.
- [43] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, Qiaozhu Mei. LINE: Large-scale Information Network Embedding. WWW '15: Proceedings of the 24th International Conference on World Wide Web-May 2015 Pages 1067–1077.
- [44] Tao-yang Fu, Wang-Chien Lee, Zhen Lei. HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning. CIKM '17: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Pages 1797–1806, November 2017.
- [45] Yuxiao Dong, Nitesh V. Chawla, Ananthram Swami. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. KDD '17: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 135–144, August 2017.
- [46] Eneko Agirre, Oier López de Lacalle, Aitor Soroa. Random Walks for Knowledge-Based Word Sense Disambiguation. Computational Linguistics 40 (1): 57–84, 2014.

- [47] Hui Zhang, Da-Gao Duan, Qiu-Li Zhang. RWRel: A fast training framework for random walk-based knowledge graph embedding. ACAI '21: Proceedings of the 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence, Article No.: 67, Pages 1–6, December 2021.
- [48] Ali Assia, Wajdi Dhiflib. Instance Matching in Knowledge Graphs through Random Walks and Semantics. Future Generation Computer Systems Volume 123, Pages 73-84, October 2021.
- [49] Qiao Liu, Liuyi Jiang, Minghao Han, Yao Liu, Zhiguang Qin. Hierarchical Random Walk Inference in Knowledge Graphs. SIGIR '16: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pages 445–454, 7 July 2016
- [50] Maximilian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. 31st Conference on Neural Information Processing Systems (NIPS 2017)
- [51] Yongqi Zhang, Quanming Yao, and James T. Kwok. Bilinear Scoring Function Search for Knowledge Graph Learning. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 45, NO. 2, FEBRUARY 2023.
- [52] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Transactions on Knowledge and Data Engineering, 1041-4347 (c) 2017
- [53] Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), Vol. 30 No. 1, 2016.
- [54] Shizhu He, Kang Liu, Guoliang Ji Jun Zh. Learning to Represent Knowledge Graphs with Gaussian Embedding. CIKM '15: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management Pages 623–632, October 2015.
- [55] Zhiyuan Liu, Yankai Lin, Maosong Sun. World Knowledge Representation. Representation Learning for Natural Language Processing pp 163–216,04 July 2020.

- [56] Han Xiao, Minlie Huang, Xiaoyan Zhu. TransG : A Generative Model for Knowledge Graph Embedding. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) 2316–2325, August 2016.
- [57] 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 International Conference on Machine Learning, Bellevue,WA, USA, 28 June–2 July 2011; pp. 809–816.
- [58] Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
- [59] Chengjin Xu, Mojtaba Nayyeri, Yung-Yu Chen, Jens Lehmann. Knowledge Graph Embeddings in Geometric Algebras. arXiv:2010.00989v4 [cs.LG] 22 Mar 2021.
- [60] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, “A semantic matching energy function for learning with multi-relational data,” *Mach. Learn.*, vol. 94, no. 2, pp. 233–259, 2014.
- [61] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, “Reasoning with neural tensor networks for knowledge base completion,” in *Adv. Neural Inf. Process. Syst.*, 2013, pp. 926–934.
- [62] Q. Liu, H. Jiang, A. Evdokimov, Z.-H. Ling, X. Zhu, S. Wei, and Y. Hu, “Probabilistic reasoning via deep learning: Neural association models,” in arXiv:1603.07704, 2016.
- [63] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, Philip S. Yu. A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs. *IEEE Access*, Vol. 10 pp. 75729 - 75741, 22 July 2022.
- [64] 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 European SemanticWeb Conference, Anissaras, Greece, 3–7 June 2018; pp. 593–607.

- [65] Thomas N. Kipf, Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907v4 [cs.LG] 22 Feb 2017.
- [66] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, Tat-Seng Chua. KGAT: Knowledge Graph Attention Network for Recommendation. arXiv:1905.07854v2 [cs.LG] 8 Jun 2019.
- [67] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. arXiv:1707.01476v6 [cs.LG] 4 Jul 2018.
- [68] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Martinata, Paolo Merialdo and DENILSON BARBOSA. Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. ACM Transactions on Knowledge Discovery from Data, Volume 15, Issue 2, Article No.: 14pp 1–49, 2021.
- [69] Cao, J., Fang, J., Meng, Z., Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. ACM Computing Surveys, 56(6), 1-42.
- [70] Jiahang CAO, Jinyuan Fang, Zaiqiao meng and Shangsong Liang. Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces. ACM Computing Surveys, Volume 56, Issue 6, Article No.: 159pp 1–42, 13 march 2024.
- [71] Hongbo Liu, Yue Chen, Peng He, Chao Zhang, Hao Wu and Jiange Zhang. An inductive knowledge graph embedding via combination of subgraph and type information. Scientific Reports, 13:21228, 2023.
- [72] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A Survey on Knowledge Graph-Based Recommender Systems. IEEE transactions on knowledge and data engineering VOL. 34, NO. 8, 2022.
- [73] Xiaohan Zou. A Survey on Application of Knowledge Graph, IOP Conf. Series: Journal of Physics: Conf. Series 1487 (2020) 012016.
- [74] Kevin McCoy, Sateesh Gudapati, Lawrence He, Elaina Horlander, David Kartchner, Soham Kulkarni , Nidhi Mehra, Jayant Prakash, Helena

- Thenot, Sri Vivek Vanga, Abigail Wagner, Brandon White and Cassie S. Mitchell. Biomedical Text Link Prediction for Drug Discovery: A Case Study with COVID-19. *Pharmaceutics* 2021, 13, 794.
- [75] Z. Liu, Z.-Y. Niu, H. Wu, and H. Wang. Knowledge aware conversation generation with explainable reasoning over augmented graphs, in *EMNLP*, 2019, pp. 1782–1792.
- [76] S. Moon, P. Shah, A. Kumar, and R. Subba, OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs,” in *ACL*, 2019, pp. 845–854.
- [77] Stephen Bonner, Ian P. Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, Charles Tapley Hoyt, William L. Hamilton. Understanding the performance of knowledge graph embeddings in drug discovery, *Artificial Intelligence in the Life Sciences* 2 (2022) 100036.
- [78] JOUPPI, Norman P., YOUNG, Cliff, PATIL, Nishant, et al. In-datacenter performance analysis of a tensor processing unit. In : *Proceedings of the 44th annual international symposium on computer architecture*. 2017. p. 1-12.
- [79] GUL, Beenish, KHAN, Imran Ali, MUSTAFA, Saad, et al. CPU and RAM energy-based SLA-aware workload consolidation techniques for clouds. *IEEE Access*, 2020, vol. 8, p. 62990-63003.
- [80] Kaushik Dutta and Debra Vandermeer. A Case for Considering Energy Consumption in Software Development and Architecture Decisions. *arXiv:2311.01680 [cs.SE]*, Nov 2023.
- [81] RADOVANOVIĆ, Ana, KONINGSTEIN, Ross, SCHNEIDER, Ian, et al. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems*, 2022, vol. 38, no 2, p. 1270-1280.
- [82] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *KDD*, page 353362, 2016.
- [83] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach, in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, 2014, pp. 283–292.

- [84] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma. Collaborative knowledge base embedding for recommender systems. In Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 353–362.
- [85] Dalton, J., Dietz, L., Allan, J.. Entity query feature expansion using knowledge base links, Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. pp. 365–374. ACM 2014.
- [86] Raviv, H., Kurland, O., Carmel, D.. Document retrieval using entity-based language models. Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 65–74. ACM 2016.
- [87] R. T. Sousa, S. Silva, and C. Pesquita, Evolving knowledge graph similarity for supervised learning in complex biomedical domains. BMC Bioinformatics, vol. 21, no. 1, p. 6, 2020.
- [88] S. K. Mohamed, V. Nováček, and A. Nounu, “Discovering protein drug targets using knowledge graph embeddings,” Bioinformatics, vol. 36, no. 2, pp. 603–610, 2020.
- [89] X. Lin, Z. Quan, Z.-J. Wang, T. Ma, and X. Zeng, “KGNN: Knowledge graph neural network for drug-drug interaction prediction.” IJCAI, 2020.
- [90] Xiou Ge, Yun-Cheng Wang, Bin Wang. Knowledge graph embedding: an overview , arXiv:2309.12501v1 [cs.AI] 21 Sep 2023.
- [91] Dat Quoc Nguyen. A survey of embedding models of entities and relationships for knowledge graph completion, Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs), pages 1–14 Barcelona, Spain (Online), December 13, 2020.
- [92] Mouloud Iferroudjene, Victor Charpenay and Antoine Zimmermann. FB15k-CVT: A Challenging Dataset for Knowledge Graph Embedding Models, 17th International Workshop on Neural-Symbolic Learning and Reasoning, Certosa di Pontignano, Siena, Italy 2023.

- [93] Bollacker, K.D.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge; SIGMOD: Vancouver, BC, Canada, 2008; pp. 1247–1250.
- [94] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs, arXiv:2005.00687v7 [cs.LG] 25 Feb 2021.
- [95] Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D. Mendes, P. N.; Hellmann, S.; Morsey, M.; Kleef, P. V.; Auer, S.; et al. DBpedia—A Large-Scale, Multilingual Knowledge base Extracted from Wikipedia; Springer: Berlin/Heidelberg, Germany, 2015; Volume 6, pp. 167–195.
- [96] Mahdisoltani, F.; Biega, J.A.; Suchanek, F.M. YAGO3: A Knowledge Base from MultilingualWikipedias. In Proceedings of the CIDR, Asilomar, CA, USA, 4–7 January 2015.
- [97] Wenjie Zheng, Wenxue Wang<sup>†</sup>, Fulan Qian, Shu Zhao and Yanping Zhang, Hyperbolic Hierarchical Knowledge Graph Embeddings for Link Prediction in Low Dimensions, arXiv:2204.13704v1 [cs.LG] 28 Apr 2022.
- [98] Qianlong Wen, Zhongyu Ouyang, Chunhui Zhang, Yiyue Qian, Yanfang Ye, Chuxu Zhang. Contrastive Graph Representation Learning with Cross-view Reconstruction. ICLR 2023 Conference Withdrawn Submission, 2023.
- [99] Kamrul Islam, Sabeur Aridhi, Malika Smaïl-Tabbone. Negative Sampling and Rule Mining for Explainable Link Prediction in Knowledge Graphs. Knowledge-Based Systems, 2022, 250, pp.109083.
- [100] NICKEL, Maximilian, MURPHY, Kevin, TRESP, Volker, et al. A review of relational machine learning for knowledge graphs. Proceedings of the IEEE, 2015, vol. 104, no 1, p. 11-33.
- [101] Shima Khoshraftar, Aijun An.A Survey on Graph Representation Learning Methods. ACM Transactions on Intelligent Systems and Technology Vol. 15, No. 1, 2024.

- [102] Zeinab Shokrzadeh, Mohammad-Reza Feizi-Derakhshi, Mohammad-Ali Balafar, Jamshid Bagherzadeh Mohasefi. Knowledge graph-based recommendation system enhanced by neural collaborative filtering and knowledge graph embedding. *Ain Shams Engineering Journal* 15 (2024) 102263.
- [103] Genet Asefa Gesese, Russa Biswas, and Harald Sack. A Comprehensive Survey of Knowledge Graph Embeddings with Literals: Techniques and Applications. *arXiv:1910.12507*, 2020.
- [104] Dat Quoc Nguyen. 2020. A survey of embedding models of entities and relationships for knowledge graph completion. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 1–14, Barcelona, Spain (Online). Association for Computational Linguistics.
- [105] Komal K. Teru 1 2 Etienne G. Denis 1 2 William L. Hamilton. Inductive Relation Prediction by Subgraph Reasoning. *Proceedings of the 37 th International Conference on Machine Learning, Vienna, Austria, PMLR* 119, 2020.
- [106] Hongbo Liu, Yue Chen, Peng He, Chao Zhang, Hao Wu & Jiange Zhang. An inductive knowledge graph embedding via combination of subgraph and type information. *Scientific Reports* volume 13, Article number: 21228 2023.
- [107] UK Parliament POST. Energy consumption of ICT. *POSTnote. Number 677* September 2022.
- [108] Kaushik Dutta and Debra VanderMeer. A Case for Considering Energy Consumption in Software Development and Architecture Decisions. *arXiv:2311.01680 [cs.SE]* 3 Nov 2023.
- [109] MALMODIN, Jens, LÖVEHAGEN, Nina, BERGMARK, Pernilla, et al. ICT sector electricity consumption and greenhouse gas emissions–2020 outcome. *Telecommunications Policy*, 2024, vol. 48, no 3, p. 102701.
- [110] Neeraj Tiwary, Shahrul Azman Mohd Noah, Fariza Fauzi and Tan Siok Yee. Max Explainability Score–A quantitative metric for explainability

evaluation in knowledge graph-based recommendations. *Computers and Electrical Engineering* 116 (2024) 109190.

- [111] Suzuki, T., Oyama, S. and Kurihara, M. Explainable Recommendation Using Review Text and a Knowledge Graph. In *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)* Los Angeles, CA, USA, 9–12 December 2019; Institute of Electrical and Electronics Engineers Inc.: Interlaken, Switzerland, 2019; pp. 4638–4643.
- [112] Lully, V.; Laublet, P.; Stankovic, M.; Radulovic, F. Enhancing explanations in recommender systems with knowledge graphs. *Procedia Comput. Sci.* 2018, 137, 211–222.
- [113] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, “Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences,” in *Proc. World Wide Web Conf.*, 2019, pp. 151–161.
- [114] Q. Li, X. Tang, T. Wang, H. Yang, and H. Song, “Unifying task-oriented knowledge graph learning and recommendation,” *IEEE Access*, vol. 7, pp. 115 816–115 828, 2019.  
Recommender Systems. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, VOL. 34, NO. 8, AUGUST 2022.
- [115] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, “Multi-task feature learning for knowledge graph enhanced recommendation,” in *Proc. World Wide Web Conf.*, 2019, pp. 2000–2010.
- [116] X. Xin, X. He, Y. Zhang, Y. Zhang, and J. Jose, “Relational collaborative filtering: Modeling multiple item relations for recommendation,” in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 125–134.
- [117] Dongze Li, Hanbing Qu and Jiaqiang Wang. A Survey on Knowledge Graph-Based Recommender Systems. *China Automation Congress (CAC)*, 2023.
- [118] Devam Mondal and Carlo Lipizzi. Ensuring Responsible Sourcing of Large Language Model Training Data Through Knowledge Graph Comparison. *arXiv:2407.02659v1 [cs.CL]* 2 Jul 2024.

- [119] YZHANG, Yifei, WANG, Xintao, LIANG, Jiaqing, et al. Chain-of-Knowledge: Integrating Knowledge Reasoning into Large Language Models by Learning from Knowledge Graphs. arXiv preprint arXiv:2407.00653, 2024.
- [120] Joao T. Aparicio, Elisabete Arsenio, Francisco Santos and Rui Henriques. Using dynamic knowledge graphs to detect emerging communities of knowledge. *Knowledge-Based Systems*, Volume 294, 21, 2024, 111671.
- [121] Xiangru Zhu, Zhixu Li, Xiaodan Wang, Xueyao Jiang, Penglei Sun, Xuwu Wang, Yanghua Xiao and Nicholas Jing Yuan. Multi-Modal Knowledge Graph Construction and Application: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, Volume 36, Issue 2 Feb. 2024, pp 715–735.
- [122] Wanying Liang, Pasquale De Meo, Yong Tang, Jia Zhu. A Survey of Multi-modal Knowledge Graphs: Technologies and Trends. *ACM Computing Surveys* Accepted on April 2024.