APSIPA Transactions on Signal and Information Processing, 2024, 13, e1

Overview Paper

Knowledge Graph Embedding: An Overview

Xiou Ge[1*], Yun Cheng Wang[1], Bin Wang[2] and C.-C. Jay Kuo[1]

[1] *University of Southern California, Los Angeles, CA, USA*
[2] *Institute for Infocomm Research (I2R), A*STAR, Singapore*

ABSTRACT

Many mathematical models have been leveraged to design embeddings for representing Knowledge Graph (KG) entities and relations for link prediction and many downstream tasks. These mathematically-inspired models are not only highly scalable for inference in large KGs, but also have many explainable advantages in modeling different relation patterns that can be validated through both formal proofs and empirical results. In this paper, we make a comprehensive overview of the current state of research in KG completion. In particular, we focus on two main branches of KG embedding (KGE) design: 1) distance-based methods and 2) semantic matching-based methods. We discover the connections between recently proposed models and present an underlying trend that might help researchers invent novel and more effective models. Next, we delve into CompoundE and CompoundE3D, which draw inspiration from 2D and 3D affine operations, respectively. They encompass a broad spectrum of distance-based embedding techniques. We will also discuss an emerging approach for KG completion which leverages pre-trained language models (PLMs) and textual descriptions of entities and relations and offer insights into the integration of KGE embedding methods with PLMs for KG completion.

*Corresponding author: Xiou Ge, xiouge@usc.edu.

20 September 2023; Revised 02 January 2024
ISSN 2048-7703; DOI 10.1561/116.00000065

Keywords: Knowledge Graph Embedding, Knowledge Graph Completion, Link Prediction.

1 Introduction

Knowledge Graphs (KGs) serve as vital repositories of information for many real-world applications and services, including search engines, virtual assistants, knowledge discovery, and fraud detection. The construction of KGs primarily involves domain expert curation or the automated extraction of data from vast web corpora. Despite the precision achieved by machine learning models in entity and relation extraction, they can introduce errors during KG construction. Furthermore, due to the inherent incompleteness of entity information, KG embedding (KGE) techniques come into play to identify missing relationships between entities. Over the past decade, there has been a surge in interest with the creation of various KGE models as evidenced in Figure 1. As such, it will be valuable to have an overview of extant KGE models to compare their similarities and differences, as well as a summary of research resources such as public KGs, benchmarking datasets, and leaderboards. In this paper, we will give a comprehensive overview of previous developments in KGE models. In particular, we will focus on distance-based and semantic matching KGE models. In recent development of KGE models, we have observed an interesting trend of combining different geometric transformations to improve the performance of existing KGE models. Basic transformations, including translation, rotation, scaling, reflection, and shear, are simple yet very powerful tools for representing relations between entities in KG. In this paper, we will also present how these tools can be combined to come up with more powerful models.

1.1 Background

KG finds vast and diverse applications. It enables swift retrieval of structured data about target entities during user searches. For instance, when you search for a well-known person, place, or popular topic on Google, the Google Knowledge Panel, shown in Figure 2, accompanies search results, providing quick insights into the subject of interest. The data source for the Knowledge Panel is the Google KG, launched in 2012, initially derived from Freebase, an open-source KG acquired by Google in 2010, and later augmented with data from sources like Wikidata.

KG information integration extends to various domains, with E-commerce companies creating user and product KGs and merging them with other KGs to gain business intelligence. Hospitals and clinics employ KGs to share patient

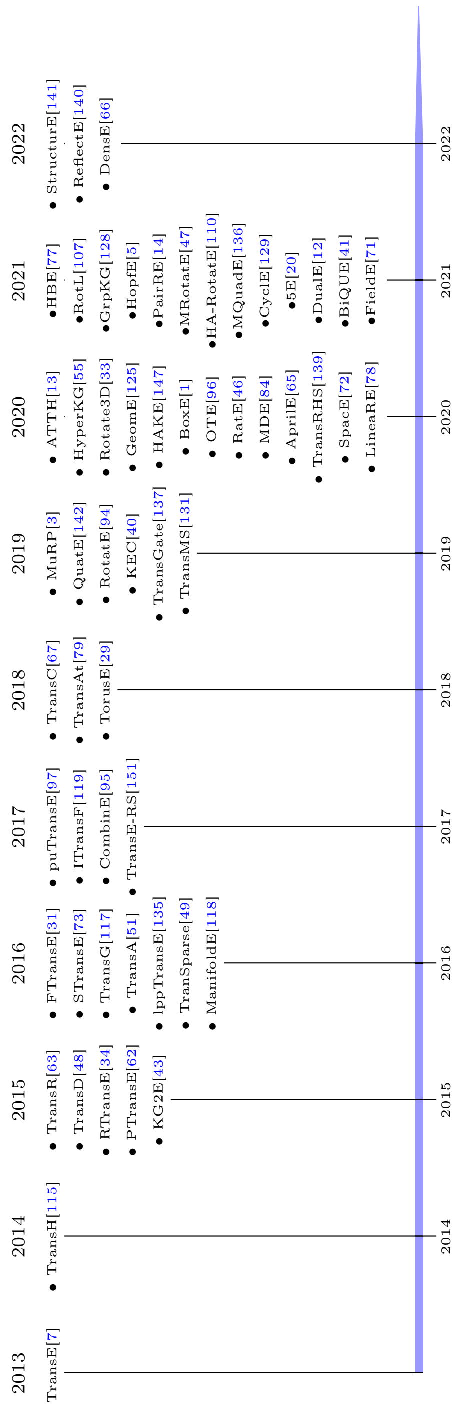


Figure 1: Timeline of Knowledge Graph Embedding models.

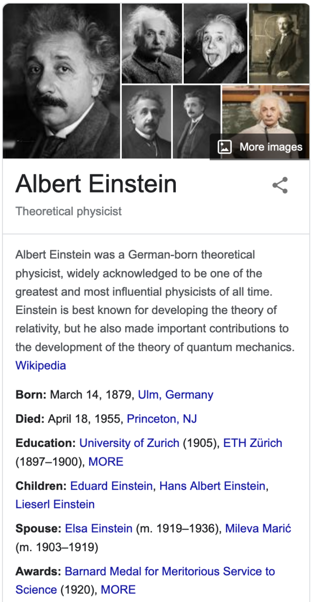


Figure 2: Illustration of Knowledge Panel from Google Search.

medical conditions, especially for patients relocating to different areas. Financial institutions use KGs to track illegal activities such as money laundering. Moreover, KGs serve as essential information sources for AI-powered virtual assistants like Siri, Alexa, and Google Assistant. Natural Language Understanding (NLU) algorithms analyze dialogs, extracting keywords to locate relevant KG subgraphs. By traversing these subgraphs, the assistants generate sensible responses using Natural Language Generation models. KGs also find applications in music recommendation systems, event forecasting based on temporal KGs, and more.

KG representation learning has been a subject of intensive research in recent years and remains a foundational challenge in Artificial Intelligence (AI) and Data Engineering. KGs are composed of triples, denoted as (h, r, t), where h and t denote head and tail entities, while r signifies the connecting relation. For instance, the statement "Los Angeles is located in the USA" is encapsulated as the triple (Los Angeles, **isLocatedIn**, USA). KGE plays a critical role in a range of downstream applications, including multihop reasoning [27, 89], KG alignment [18, 35, 36], entity classification [39, 113].

The evaluation of KGE models often revolves around the link prediction task, assessing their ability to predict t given h and r, or h given r and t. The effectiveness of KGE models is determined by how closely their predictions align with the ground truth. Designing effective KGE models presents several challenges. First, KGE needs to be scalable since real-world KGs often contain millions of entities and relations. Designing embeddings that scale efficiently to handle large graphs is a significant challenge. Second, KGs are typically incomplete and subject to continuous updates. It is desirable for embedding models for handling missing data and capturing temporal dynamics and the history of relationships. Third, embeddings must also be expressive enough to capture the complexity of real-world relationships, such as 1-to-N, N-to-1, N-to-N, antisymmetric, transitive, and hierarchical relations and multi-modal data. Fourth, some entities and relations are rare. Embeddings should handle the long-tail distribution of data effectively.

We have collected a list of survey papers as shown in Table 1. Among these surveys, [6, 9, 11, 19, 50, 74, 83, 109] focus on discussing different embedding models, whereas [45, 50, 98] discuss the use of KG for reasoning and different applications. [42] elucidates the evolution of KGs and the reasons for their invention in a historical context. [45] summarizes methods for the creation, enrichment, quality assessment, refinement, and publication of KGs, and provides an overview of prominent open KGs and enterprise KGs. [98] also discusses the advantages and disadvantages of using KGs as background knowledge in the context of Explainable Machine Learning. However, none of these papers discuss the intrinsic connections between different distance-based embedding models that use geometric transformations. We believe that this paper will be helpful to the research community by providing a unique perspective on this topic.

1.2 Our Contributions

Our main contributions in this paper can be summarized as follows.

- We review different KGE models, focusing on distance-based and semantic matching models.
- We collect relevant resources for KG research, including previously published survey papers, major open sourced KGs, and KG benchmarking datasets for link prediction, as well as link prediction performance on some of the most popular datasets.
- We discover the connections between recently published KGE models and propose CompoundE and CompoundE3D, which follow this direction of thought.
- We discuss recent work that leverages neural network models such as graph neural networks and pretrained language models, and how embedding-based models can be combined with neural network-based models.

Table 1: Survey Papers.

No.	Year	Title	Author	Venue
1	2013	Representation Learning: A Review and New Perspectives [6]	Yoshua Bengio, Aaron C. Courville, Pascal Vincent.	IEEE Transactions on Pattern Analysis and Machine Intelligence
2	2016	A Review of Relational Machine Learning for Knowledge Graphs [74]	Maximilian Nickel, Kevin Murphy, Volker Tresp, Evgeniy Gabrilovich.	Proceedings of the IEEE
3	2017	Knowledge Graph Embedding: A Survey of Approaches and Applications [109]	Quan Wang, Zhendong Mao, Bin Wang, Li Guo.	IEEE Transactions on Knowledge and Data Engineering
4	2018	A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. [9]	HongYun Cai, Vincent W. Zheng, Kevin Chen-Chuan Chang.	IEEE Transactions on Knowledge and Data Engineering
5	2020	A review: Knowledge Reasoning over Knowledge Graph. [19]	Xiaojun Chen, Shengbin Jia, Yang Xiang.	Expert Systems with Applications
6	2021	Knowledge Graphs [42]	Claudio Gutierrez, Juan F. Sequeda.	Communications of the ACM
7	2021	Knowledge Graph Embedding for Link Prediction: A Comparative Analysis [83]	Rossi, Andrea and Barbosa, Denilson and Firmani, Donatella and Matinata, Antonio and Merialdo, Paolo.	ACM Transactions on Knowledge Discovery from Data
8	2021	Knowledge Graphs [45]	Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid,Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, Antoine Zimmermann.	ACM Computing Surveys

Table 1: Continued.

No.	Year	Title	Author	Venue
9	2022	Knowledge Graphs: A Practical Review of the Research Landscape [54]	Mayank Kejriwal.	Information
10	2022	Knowledge Graphs as Tools for Explainable Machine Learning: A survey [98]	Ilaria Tiddi, Stefan Schlobach.	Artificial Intelligence
11	2022	A Survey on Knowledge Graphs: Representation, Acquisition, and Applications [50]	Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, Philip S. Yu.	IEEE Transactions on Neural Networks and Learning Systems
12	2022	Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces [11]	Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, Shangsong Liang.	arXiv prepints

The rest of this paper is organized as follows. In Section 2, we first introduce existing KGE models in both distance-based and semantic matching-based categories. We also discuss a number of commonly used loss functions and their suitability for different types of KGE scoring functions. In Section 3, we present CompoundE, followed by CompoundE3D, which unifies all distance-based KGE models that use affine operations. In Section 4, we summarize a list of open sourced KGs, popular benchmarking datasets for performance evaluation, and the evaluation metrics used in these datasets. We also provide the recent leaderboard for some popular datasets. In Section 5, we discuss existing neural network-based models for KG completion and emerging directions that use pretrained language models. Finally, in Section 6, we make concluding remarks.

2 Existing Models

KGE models are often categorized based on scoring functions and tools applied to model entity-relation interactions and representations. In this paper, we mainly discuss two major classes, namely 1) distance-based models, and 2) semantic matching models.

2.1 Distance-based Models

Distance-based scoring function is one of the most popular strategies for learning KGE. The intuition behind this strategy is that relations are modeled as transformations to place head entity vectors in the proximity of their corresponding tail entity vectors or vice versa. For a given triple (h, r, t), the goal is to minimize the distance between h and t vectors after the transformation introduced by r.

TransE [7] is one of the first KGE models that interpret relations between entities as translation operations in vector space. Let $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ denote the embedding for head, relation, and tail of a triple, respectively. TransE scoring function is defined as:

$$f_r(h, r) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p, \tag{1}$$

where $p = 1$ or 2 denote 1-Norm or 2-Norm, respectively. However, this efficient model has difficulty modeling complex relations such as 1-N, N-1, N-N, symmetric and transitive relations. Many later works attempt to overcome this shortcoming. For example, TransH [115] projects entity embedding onto relation-specific hyperplanes so that complex relations can be modeled by the translation embedding model. Formally, let $\mathbf{w}_r$ be the normal vector to a relation-specific hyperplane, then the head and tail representation in the hyperplane can be written as,

Table 2: Distance-based KGE models

Model	Ent. emb.	Rel. emb.	Scoring Function	Space
TransE [7]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\lVert\mathbf{h} + \mathbf{r} - \mathbf{t}\rVert_{1/2}$	$O(md + nd)$
TransR [63]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^k, \mathbf{M}_r \in \mathbb{R}^{k\times d}$	$-\lVert\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}\rVert_2^2$	$O(mdk + nd)$
TransH [115]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{w}_r \in \mathbb{R}^d$	$-\left\lVert\left(\mathbf{h} - \mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r\right) + \mathbf{r} - \left(\mathbf{t} - \mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r\right)\right\rVert_2^2$	$O(md + nd)$
TransA [51]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d, \mathbf{M}_r \in \mathbb{R}^{d\times d}$	$(\lvert\mathbf{h} + \mathbf{r} - \mathbf{t}\rvert)^\top \mathbf{W_r}(\lvert\mathbf{h} + \mathbf{r} - \mathbf{t}\rvert)$	$O(md^2 + nd)$
TransF [31]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$(\mathbf{h} + \mathbf{r})^\top \mathbf{t} + (\mathbf{t} - \mathbf{r})^\top \mathbf{h}$	$O(md + nd)$
TransD [48]	$\mathbf{h}, \mathbf{h}_p, \mathbf{t}, \mathbf{t}_p \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{r}_p \in \mathbb{R}^k$	$-\left\lVert\left(\mathbf{r}_p\mathbf{h}_p^T + \mathbf{I}\right)\mathbf{h} + \mathbf{r} - \left(\mathbf{r}_p\mathbf{t}_p^T + \mathbf{I}\right)\mathbf{t}\right\rVert_2^2$	$O(mk + nd)$
TransM [30]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d,\ w_r \in \mathbb{R}$	$-w_r\lVert\mathbf{h} + \mathbf{r} - \mathbf{t}\rVert_{1/2}$	$O(md + nd)$
TranSparse [49]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^k, \mathbf{M}_r(\theta_r) \in \mathbb{R}^{k\times d}$	$-\lVert\mathbf{M}_r(\theta_r)\mathbf{h} + \mathbf{r} - \mathbf{M}_r(\theta_r)\mathbf{t}\rVert_{1/2}^2$	$O(mdk + nd)$
		$\mathbf{M}_r^1\left(\theta_r^1\right), \mathbf{M}_r^2\left(\theta_r^2\right) \in \mathbb{R}^{k\times d}$	$-\left\lVert\mathbf{M}_r^1\left(\theta_r^1\right)\mathbf{h} + \mathbf{r} - \mathbf{M}_r^2\left(\theta_r^2\right)\mathbf{t}\right\rVert_{1/2}^2$	
ManifoldE [118]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\left\lVert\mathcal{M}(h, r, t) - D_r^2\right\rVert^2$	$O(md + nd)$
TorusE [29]	$[\mathbf{h}], [\mathbf{t}] \in \mathbb{T}^n$	$[\mathbf{r}] \in \mathbb{T}^n$	$\min_{(x,y)\in([h]+[r])\times[t]} \lVert x - y\rVert_i$	$O(md + nd)$
RotatE [94]	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$-\lVert\mathbf{h} \circ \mathbf{r} - \mathbf{t}\rVert$	$O(md + nd)$
PairRE [14]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r^H}, \mathbf{r^T} \in \mathbb{R}^d$	$-\lVert\mathbf{h} \circ \mathbf{r}^H - \mathbf{t} \circ \mathbf{r}^T\rVert$	$O(md + nd)$

Table 3: Semantic matching-based KGE models.

Model	Ent. emb.	Rel. emb.	Scoring Function	Space
RESCAL [76]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^{d \times d}$	$\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$	$O(md^2 + nd)$
DistMult [127]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\mathbf{h}^\top \operatorname{diag}(\mathbf{r}) \mathbf{t}$	$O(md + nd)$
HolE [75]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\mathbf{r}^\top (h \star t)$	$O(md + nd)$
ANALOGY [64]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	${}^1\hat{\mathbf{M}}_r \in \mathbb{R}^{d \times d}$	$\mathbf{h}^\top \hat{\mathbf{M}}_r \mathbf{t}$	$O(md + nd)$
ComplEx [101]	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$\operatorname{Re}\left(\langle \mathbf{r}, \mathbf{h}, \bar{\mathbf{t}} \rangle\right)$	$O(md + nd)$
SimplE [53]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{r}' \in \mathbb{R}^d$	$\frac{1}{2}\left(\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle + \langle \mathbf{t}, \mathbf{r}', \mathbf{h} \rangle\right)$	$O(md + nd)$
Dihedral [123]	$\mathbf{h}^{(l)}, \mathbf{t}^{(l)} \in \mathbb{R}^2$	$\mathbf{R}^{(l)} \in \mathbb{D}_K$	$\sum_{l=1}^{L} \mathbf{h}^{(l)\top} \mathbf{R}^{(l)} \mathbf{t}^{(l)}$	$O(md + nd)$
TuckER [4]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}_e^d$	$\mathbf{r} \in \mathbb{R}_r^d$	$\mathcal{W} \times_1 \mathbf{h} \times_2 \mathbf{r} \times_3 \mathbf{t}$	$O(d_r d_e^2 + md_r + nd_e)$
QuatE [142]	$Q_h, Q_t \in \mathbb{H}^d$	$W_r \in \mathbb{H}^d$	$Q_h \otimes W_r^{\triangleleft} \cdot Q_t$	$O(md + nd)$
DualE [12]	$Q_h, Q_t \in \mathbb{H}^d$	$W_r, T_r \in \mathbb{H}^d$	$(Q_h \otimes W_r^{\triangleleft} + T_r) \cdot Q_t$	$O(md + nd)$
CrossE [143]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\sigma\left(\tanh\left(\mathbf{c}_r \circ \mathbf{h} + \mathbf{c}_r \circ \mathbf{h} \circ \mathbf{r} + \mathbf{b}\right) \mathbf{t}^\top\right)$	$O(md + nd)$
SEEK [126]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\sum s_{x,y} \left\langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_{w_{x,y}} \right\rangle$	$O(md + nd)$

$$\mathbf{h}_{\perp} = \mathbf{h} - \mathbf{w}_r^{\top}\mathbf{h}\mathbf{w}_r, \quad \mathbf{t}_{\perp} = \mathbf{t} - \mathbf{w}_r^{\top}\mathbf{t}\mathbf{w}_r. \tag{2}$$

The projected representations are then linked together using the same translation relationship,

$$f_r(h, r) = \|\mathbf{h}_{\perp} + \mathbf{r} - \mathbf{t}_{\perp}\|_2^2. \tag{3}$$

However, this orthogonal projection prevents the model from encoding inverse and composition relations. A similar idea called TransR [63] transforms entities into a relation-specific space instead. The TransR scoring function can be written as,

$$f_r(h, r) = \|\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}\|_2^2. \tag{4}$$

However, the relation-specific transformation introduced in TransR requires $O(kd)$ additional parameters. To save the additional parameters introduced, TransD [48] uses entity projection vectors to populate the mapping matrices, instead of using a dense matrix. TransD reduces the additional parameters from $O(kd)$ to $O(k)$. The scoring function can be written as,

$$f_r(h, r) = \left\| \left(\mathbf{r}_p\mathbf{h}_p^{\top} + \mathbf{I}\right)\mathbf{h} + \mathbf{r} - \left(\mathbf{r}_p\mathbf{t}_p^{\top} + \mathbf{I}\right)\mathbf{t} \right\|_2^2. \tag{5}$$

With the same goal of saving additional parameters, TranSparse enforces the transformation matrix to be a sparse matrix. The scoring function can be written as,

$$f_r(h, t) = \|\mathbf{M}_r\left(\theta_r\right)\mathbf{h} + \mathbf{r} - \mathbf{M}_r\left(\theta_r\right)\mathbf{t}\|_{1/2}^2, \tag{6}$$

where $\theta_r \in [0, 1]$ is the sparse degree for the mapping matrix $\mathbf{M}_r$. Variants of TranSparse [49] include separate mapping matrices for head and tail. TransM [30] assigns different weights to complex relations for better encoding power. TransMS [131] attempts to consider multidirectional semantics using nonlinear functions and linear bias vectors. TransF [31] mitigates the burden of relation projection by explicitly modeling the basis of projection matrices. ITransF [119] makes use of concept projection matrices and sparse attention vectors to discover hidden concepts within relations.

In recent years, researchers expand their focus to spaces other than Euclidean geometry. TorusE [28] projects embedding in an n-dimensional torus space, where $[\mathbf{h}], [\mathbf{r}], [\mathbf{t}] \in \mathbb{T}^n$ denotes the projected representation of head, relation, tail. TorusE models relational translation in Torus space by optimizing the objective as follows.

$$\min_{(\mathbf{x},\mathbf{y})\in([\mathbf{h}]+[\mathbf{r}])\times[\mathbf{t}]} \|\mathbf{x} - \mathbf{y}\|_i. \tag{7}$$

Multi-Relational Poincaré model (MuRP) [3] embeds KG entities in a Poincaré ball of hyperbolic space. It transforms entity embeddings using relation-specific Möbius matrix-vector multiplication and Möbius addition.

The negative curvature introduced by hyperbolic space is empirically better in capturing the hierarchical structure in KGs. However, MuRP has difficulty encoding relation patterns and only uses a constant curvature. ROTH [13] improve over MuRP by introducing a relation-specific curvature.

RotatE [94] models entities in the complex vector space and interprets relations as rotations instead of translations. Formally, let $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^d$ denote the representation of head, relation, and tail of a triple in the complex vector space. The RotatE scoring function can be defined as,

$$f_r(h,t) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\| . \tag{8}$$

The self-adversarial negative sampling strategy also contributes to RotatE's significant performance improvement compared to its predecessors. Quite a few models attempt to extend RotatE. MRotatE adds an entity rotation constraint to the optimization objective to handle multifold relations. HAKE rewrites the rotation formula in polar coordinates and separates the scoring function into two components, that is, the phase component and the modulus component. The scoring function of HAKE can be written as,

$$f_r(h,t) = d_{r,m}(\mathbf{h}, \mathbf{t}) + \lambda d_{r,p}(\mathbf{h}, \mathbf{t}), \tag{9}$$

where

$$d_{r,p}(\mathbf{h}, \mathbf{t}) = \| \sin((\mathbf{h}_p + \mathbf{r}_p - \mathbf{t}_p)/2)\|_1, \tag{10}$$

and

$$d_{r,m}(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_m \circ ((\mathbf{r}_m + \mathbf{r}'_m)/(1 - \mathbf{r}'_m)) - \mathbf{t}_m\|_2. \tag{11}$$

This modification leads to better modeling capability of hierarchy structures in KG. Rotate3D performs quaternion rotation in 3D space and enables the model to encode non-commutative relations. Rot-Pro extends the RotatE by transforming entity embeddings using an orthogonal projection that is also idempotent. This change enables RotPro to model transitive relations. PairRE also tries to improve over RotatE. Instead of rotating the head to match the tail, PairRE [14] performs transformations on both head and tail. The scoring function can be defined as,

$$f_r(h,t) = \|\mathbf{h} \circ \mathbf{r}^{\mathbf{H}} - \mathbf{t} \circ \mathbf{r}^{\mathbf{T}}\|, \tag{12}$$

where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ are head and tail entity embedding, and $\mathbf{r}^{\mathbf{H}}, \mathbf{r}^{\mathbf{T}} \in \mathbb{R}^d$ are relation-specific weight vectors for head and tail vectors respectively, and $\circ$ is an elementwise product. In fact, this elementwise multiplication is simply the scaling operation. One advantage of PairRE compared to previous models is that it is capable of modeling subrelation structures in KG. LinearRE [78] is a similar model but adds a translation component between the scaled head and tail embedding. The transformation strategy can still be effective by adding it to entity embedding involved in relation rotation. SFBR [61] introduces a semantic filter which includes a scaling and shift component. HousE [59] and ReflectE [140] models relation as Householder reflection. UltraE [121] unifies Euclidean and hyperbolic geometry by modeling each relation as a pseudo-orthogonal transformation that preserves the Riemannian bilinear form. On the other hand, RotL [107] investigates the necessity of introducing hyperbolic geometry in learning KGE and proposes two more efficient Euclidean space KGE while retaining the advantage of flexible normalization.

2.2 Semantic Matching Models

Another related idea of developing KGE models is to measure the semantic matching score. RESCAL [76] adopts a bilinear scoring function as the objective in solving a three-way rank-r matrix factorization problem. Formally, let $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ denote the head and tail embedding and $\mathbf{M}_r \in \mathbb{R}^{d\times d}$ is the representation for relation. Then, the RESCAL scoring function can be defined as,

$$f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t}. \tag{13}$$

However, one obvious limitation of this approach is that it uses a dense matrix to represent each relation, which requires an order of magnitude more parameters compared to those using vectors. DistMult [127] reduces free parameters by enforcing the relation embedding matrix to be diagonal. Let $\mathbf{r} \in \mathbb{R}^d$ be the relation vector. Then, $\text{diag}(\mathbf{r}) \in \mathbb{R}^{d\times d}$ is the diagonal matrix constructed from $\mathbf{r}$. Then, the DistMult scoring function can be written as,

$$f_r(h,t) = \mathbf{h}^\top \text{diag}(\mathbf{r})\mathbf{t}. \tag{14}$$

However, because the diagonal matrix is symmetric, it has difficulty modeling antisymmetric relations. ANALOGY [64] has the same scoring function as RESCAL but instead it attempts to incorporate antisymmetric configurations by imposing two regularization constraints: 1) $\mathbf{M}_r\mathbf{M}_r^\top = \mathbf{M}_r^\top\mathbf{M}_r$ which requires the relation matrix to be orthonormal; 2) $\mathbf{M}_r\mathbf{M}_{r'} = \mathbf{M}_{r'}\mathbf{M}_r$ which requires the relation matrix to be commutative. HolE [75] introduces circular correlation between head and tail vectors, which can be interpreted as a compressed tensor product to capture richer interactions. The HolE scoring function can be written as,

$$f_r(h,t) = \mathbf{r}^\top (\mathbf{h} \star \mathbf{t}). \tag{15}$$

ComplEx [101] extends the bilinear product score to the complex vector space so as to model antisymmetric relations more effectively. Formally, let $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^d$ be the head, relation, tail complex vectors, and $\overline{\mathbf{t}}$ denote the complex conjugate of the $\mathbf{t}$. The ComplEx scoring function can be defined as,

$$f_r(h,t) = \text{Re}(\langle \mathbf{r}, \mathbf{h}, \overline{\mathbf{t}} \rangle). \tag{16}$$

where $\langle\cdot,\cdot,\cdot\rangle$ denotes trilinear product, and $\mathrm{Re}(\cdot)$ means taking the real part of a complex value. However, relation compositions remain difficult for ComplEx to encode. SimplE [53] models inverse of relations with an enhanced version of Canonical Polyadic decomposition. The scoring function of SimplE is defined as,

$$f_r(h,t) = \frac{1}{2}\left(\langle \mathbf{h},\mathbf{r},\mathbf{t}\rangle + \langle \mathbf{t},\mathbf{r}',\mathbf{h}\rangle\right). \tag{17}$$

TuckER [4] extends the semantic matching model to 3D tensor factorization of the binary tensor representation of KG triples. The scoring function is defined as,

$$f_r(h,t) = \mathcal{W} \times_1 \mathbf{h} \times_2 \mathbf{r} \times_3 \mathbf{t}. \tag{18}$$

with $\times_n$ indicating the tensor product along the n-th mode. QuatE [142] and DualE [12] extend from the complex representation to the hypercomplex representation with 4 degrees of freedom to gain more expressive rotational capability. Let $Q_h, W_r, Q_t \in \mathbb{H}^k$ be the representation of head, relation, and tail in quaternion space of the form $Q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{d}$. Then the QuatE scoring function is defined as,

$$f_r(h,t) = Q_h \otimes W_r^{\triangleleft} \cdot Q_t. \tag{19}$$

Specifically, the normalization of relation vector in quaternion space is defined as,

$$W_r^{\triangleleft}(p,q,u,v) = \frac{W_r}{|W_r|} = \frac{a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k}}{\sqrt{a_r^2 + b_r^2 + c_r^2 + d_r^2}}. \tag{20}$$

And the Hamiltonian product in quaternion space is computed as,

$$\begin{aligned} Q_h \otimes W_r^{\triangleleft} = {} & (a_h \circ p - b_h \circ q - c_h \circ u - d_h \circ v) \\ & + (a_h \circ q + b_h \circ p + c_h \circ v - d_h \circ u)\mathbf{i} \\ & + (a_h \circ u - b_h \circ v + c_h \circ p + d_h \circ q)\mathbf{j} \\ & + (a_h \circ v + b_h \circ u - c_h \circ q + d_h \circ p)\mathbf{k}. \end{aligned} \tag{21}$$

And the inner product in quaternion space is computed as,

$$Q_1 \cdot Q_2 = \langle a_1, a_2\rangle + \langle b_1, b_2\rangle + \langle c_1, c_2\rangle + \langle d_1, d_2\rangle. \tag{22}$$

However, one disadvantage of these models is that they require very high dimensional spaces to work well and therefore it is difficult to scale to large KGs. CrossE introduces crossover interactions to better represent the birdirectional interactions between entity and relations. The scoring function of CrossE is defined as,

$$f_r(h,t) = \sigma\left(\tanh\left(\mathbf{c}_r \circ \mathbf{h} + \mathbf{c}_r \circ \mathbf{h} \circ \mathbf{r} + \mathbf{b}\right)\mathbf{t}^\top\right), \tag{23}$$

where the relation specific interaction vector $\mathbf{c}_r$ is obtained by looking up interaction matrix $\mathbf{C} \in \mathbb{R}^{n_r \times d}$. Diehdral [123] construct elements in a dihedral group using rotation and reflection operations over a 2D symmetric polygon. The advantage of the model is with encoding relation composition. SEEK [126] and AutoSF [144] identify the underlying similarities among popular KGEs and propose an automatic framework of designing new bilinear scoring functions while also unifying many previous models. However, the search space of AutoSF is computationally intractable and it is difficult to know if one configuration will be better than another unless the model is trained and tested with the dataset. Therefore, the AutoSF search can be time-consuming. MEI [100] proposes a multi-partition embedding interaction model with block term format, to systematically control the trade-off between expressiveness and computational cost. The K local interactions in MEI essentially function as an ensemble boosting. MEIM [99] further enhances MEI by introducing core tensor to improve ensemble effects and max-rank mapping by soft orthogonality to improve model expressiveness.

2.3 Loss Functions

Loss function is an important part of KGE learning. Loss functions are designed to effectively distinguish valid triples from negative samples. The ultimate goal of optimizing the loss function is to get valid triples ranked as high as possible. In early days of KGE learning, margin-based ranking loss is widely adopted. The pairwise max-margin loss can be formally defined as,

$$L_R = \sum_{\substack{(h,r,t)\in\mathcal{G} \\ (h',r,t')\in\mathcal{G}'}} \max(0, \gamma + f_r(h,t) - f_r(h',t')), \tag{24}$$

where (h,r,t) denotes ground truth triple from the set of all valid triples $\mathcal{G}$, (h',r,t') denotes negative sample from the set of corrupted triples $\mathcal{G}'$. γ is the margin parameter which specifies how different $f_r(h,t)$ and $f_r(h',t')$ should be at optimum. In fact, a similar loss function is applied to optimize multiclass Support Vector Machine (SVM) [116]. Both distance-based embedding models, such as TransE, TransH, TransR, and TransD, and semantic matching-based models, such as LFM, NTN, and SME have successfully leveraged this scoring function. [151] proposes a Limit-based scoring loss to limit the score of positive triples so that the translation relation in positive triples can be guaranteed. The Limit-based score can be defined as,

$$L_{RS} = \sum_{\substack{(h,r,t)\in\mathcal{G} \\ (h',r,t')\in\mathcal{G}'}} \{[\gamma + f_r(h,t) - f_r(h',t')]_+ + \lambda[f_r(h,t) - \mu]_+\}. \tag{25}$$

More recently, a Double Limit Scoring Loss is proposed by [150] to independently control the golden triplets' scores and negative samples' scores. It can be defined as,

$$L_{SS} = \sum_{\substack{(h,r,t)\in\mathcal{G} \\ (h',r,t')\in\mathcal{G}'}} \{[f_r(h,t) - \mu_p]_+ + \lambda[\mu_n - f_r(h',t')]_+\}, \tag{26}$$

where $\mu_n > \mu_p > 0$. This loss function intends to encourage low distance score for positive triplets and high distance scores for negative triplets. We can also trace the usage of a similar contrastive loss from Deep Triplet Network [44] for different image classification tasks.

Self adversarial negative sampling was proposed in RotatE [94] and can be defined as,

$$L_{SANS} = -\log\sigma(\gamma - f_r(h,t)) - \sum_{i=1}^{n} p(h'_i, r, t'_i)\log\sigma(f_r(h'_i, t'_i) - \gamma). \tag{27}$$

Cross entropy or negative log-likelihood of logistic models are often used in semantic matching models where a product needs to be computed. The negative log-likelihood loss function can be defined as,

$$L_{CE} = \sum_{(h,r,t)\in\mathcal{G}\cup\mathcal{G}'} \{1 + \exp[-y_{(h,r,t)} \cdot f_r(h,t)]\}. \tag{28}$$

Binary cross entropy or Bernoulli negative log-likelihood of logistic is also a popular loss function which can be defined as,

$$L_{BCE} = \frac{1}{N_e}\sum_{i=1}^{N_e} y_i \log p_i + (1 - y_i)\log(1 - p_i). \tag{29}$$

The binary cross entropy scoring function is more suitable for neural network-based models such as ConvE. Although TuckER is a semantic matching-based model, it also uses binary cross entropy as its loss function because its implementation is similar to a neural network.

2.4 Connections between KGE Models

Another interesting finding is that we discover the connections between different recent embedding models. In recent years, there is a trend to design more effective KGE using affine transformations. In fact, models are related to each other, and many of the new embedding models emerge from models that only use the most fundamental operations, such as translation (TransE), rotation (RotatE), and scaling (PairRE). We illustrate connections between recent KGE models in Figure 3. For example, MuRE is a Euclidean version of MuRP, a popular hyperbolic space KGE model proposed in [3]. In MuRE, a diagonal matrix $\mathbf{R} \in \mathbb{R}^{d\times d}$ is applied to the head entity, and a translation vector $\mathbf{r} \in \mathbb{R}^d$ to the tail entity. The scoring function can be written as,

$$\phi(e_s, r, e_o) = -d(\mathbf{R}\mathbf{e_s}, \mathbf{e_o} + \mathbf{r})^2 + b_s + b_o, \tag{30}$$

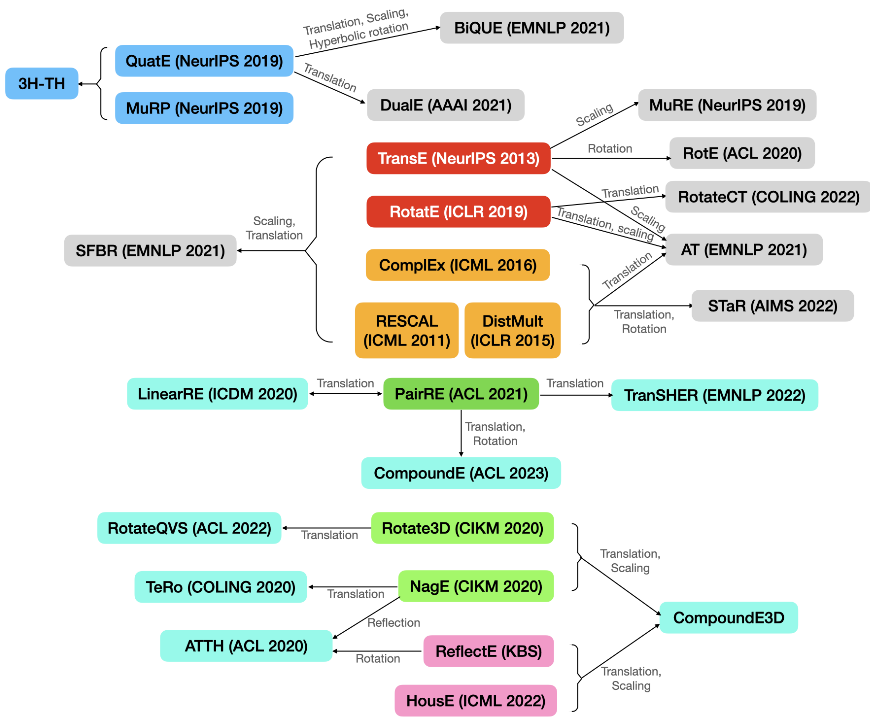


Figure 3: Connections between different knowledge graph embedding models

where b_s and b_o are the entity-specific bias terms for head and tail entities, respectively. This is essentially combining translation and scaling operations and MuRP implements a similar idea in the hyperbolic space. Similarly in [13], RotE and RotH are baseline models proposed in Euclidean space and hyperbolic space, respectively, that essentially apply 2D rotation operators to head entity embedding in the translational distance scoring function. The RotE scoring function can be defined as,

$$s(h, r, t) = d(\text{Rot}(\boldsymbol{\Theta})\mathbf{e}_h + \mathbf{r}_r, \mathbf{e}_t) + b_h + b_t \quad (31)$$

RefE and RefH can be derived similarly by applying 2D reflection operators. More recently, [130] combines translation and scaling operations with both distance-based models (TransE, RotatE) and semantic matching models (DistMult, ComplEx). Performance improvement in link prediction can be observed following this approach. SFBR [61] applies semantic filters to distance-based and semantic matching-based models. One of the more effective MLP-based filters has diagonal weight matrices, which essentially apply scaling and translation operations to the entity embeddings. STaR [58] focuses on designing bilinear product matrices of semantic matching-based models by inserting scaling and translation components in order to enable the KGE to handle complex relation types such as 1-N, N-1, and N-N relations.

Similar trend has also been observed in quaternion space KGE that was first proposed by [142]. DualE [12] introduces translation operations and combines with quaternion rotations to encode additional relation patterns, including multiplicity. BiQUE [41] further includes hyperbolic rotations and scaling operations to better model hierarchical semantics. 3H-TH adds quaternion rotation to hyperbolic embedding model MuRP to further improve link prediction performance.

Quite a few models also leverage scaling operation that is first demonstrated by PairRE [14] to have good performance in link prediction. Both LinearRE [78] and TranSHER [60] introduce translation vectors in the scaling-based scoring functions, but give different interpretations. LinearRE treats triple encoding as a linear regression problem, whereas TranSHER frames it as a translating distance model on a hyper-ellipsoid surface. Inspired by the idea of compounding affine operations for image manipulation, CompoundE [37] further includes rotation operators to encode non-commutative relation compositions. Both ReflectE [140] and HousE [59] encodes relations with householder reflections that have intrinsic symmetry property. ReflectE further explores the effect of combining translation vectors for also modeling antisymmetric relations in KG. On the other hand, HousE evaluates the effect of having a sequence of householder reflections on the link prediction performance. However, ATTH [13] was in fact the first work that introduced reflection operations in KGE. Additional operators have also been applied to existing KGEs to encode temporal information. For instance, TeRo [124] applies 2D rotation to head and tail entity embedding in TransE to encode time-specific information in temporal KG quadruples. Similarly, RotateQVS [17] adds a time-specific translation vector to entity embedding in Rotate3D [33], which leverages quaternion rotation. CompoundE3D [38] gets inspired by the evolution from RotatE to Rotate3D and proposes a KGE that unifies all geometric operators including translation, scaling, rotation, reflection, and shear in 3D space. Apart from proposing a unified framework, CompoundE3D also suggests a beam search-based procedure for finding the optimal KGE design for each KG dataset.

3 Unified Framework for Distance-based KGE with Affine Transformations: CompoundE and CompoundE3D

In this section, we will introduce the detailed formulation of CompoundE, followed by CompoundE3D. For both CompoundE and CompoundE3D, we have three forms of scoring functions, namely

- CompoundE-Head

$$f_r(h,t) = \|\mathbf{M_r} \cdot \mathbf{h} - \mathbf{t}\|, \tag{32}$$

- CompoundE-Tail

$$f_r(h,t) = \|\mathbf{h} - \hat{\mathbf{M}}_\mathbf{r} \cdot \mathbf{t}\|, \tag{33}$$

- CompoundE-Complete

$$f_r(h,t) = \|\mathbf{M_r} \cdot \mathbf{h} - \hat{\mathbf{M}}_\mathbf{r} \cdot \mathbf{t}\|, \tag{34}$$

where $\mathbf{M_r}$ and $\hat{\mathbf{M}}_\mathbf{r}$ are geometric operators to be designed. We first discuss the CompoundE operators, which include translation, scaling, and 2D rotation operations.

3.1 CompoundE Operators

First, we think it is helpful to formally introduce some definitions in group theory.

Definition 3.1. *The special orthogonal group is defined as*

$$\mathbf{SO}(n) = \left\{\mathbf{A}\middle|\mathbf{A} \in \mathbf{GL}_n(\mathbb{R}), \mathbf{A}^\top\mathbf{A} = \mathbf{I}, \det(\mathbf{A}) = 1\right\}. \tag{35}$$

Definition 3.2. *The special Euclidean group is defined as*

$$\mathbf{SE}(n) = \left\{\mathbf{A}\middle|\mathbf{A} = \begin{bmatrix}\mathbf{R} & \mathbf{v}\\ \mathbf{0} & 1\end{bmatrix}, \mathbf{R} \in \mathbf{SO}(n), \mathbf{v} \in \mathbb{R}^n\right\}. \tag{36}$$

Definition 3.3. *The affine group is defined as*

$$\mathbf{Aff}(n) = \left\{\mathbf{M}\middle|\mathbf{M} = \begin{bmatrix}\mathbf{A} & \mathbf{v}\\ \mathbf{0} & 1\end{bmatrix}, \mathbf{A} \in \mathbf{GL}_n(\mathbb{R}), \mathbf{v} \in \mathbb{R}^n\right\}. \tag{37}$$

By comparing Equations (36) and (37), we see that $\mathbf{SE}(n)$ is a subset of $\mathbf{Aff}(n)$.

Without loss of generality, consider $n = 2$. If $\mathbf{M} \in \mathbf{Aff}(2)$, we have

$$\mathbf{M} = \begin{bmatrix}\mathbf{A} & \mathbf{v}\\ \mathbf{0} & 1\end{bmatrix}, \mathbf{A} \in \mathbb{R}^{2\times 2}, \mathbf{v} \in \mathbb{R}^2. \tag{38}$$

The 2D translational matrix can be written as

$$\mathbf{T} = \begin{bmatrix}1 & 0 & v_x\\ 0 & 1 & v_y\\ 0 & 0 & 1\end{bmatrix}, \tag{39}$$

while the 2D rotational matrix can be expressed as

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{40}$$

It is easy to verify that they are both special Euclidean groups (i.e. $\mathbf{T} \in \mathbf{SE}(2)$ and $\mathbf{R} \in \mathbf{SE}(2)$). On the other hand, the 2D scaling matrix is in form of

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{41}$$

It is not a special Euclidean group but an affine group of $n = 2$ (i.e., $\mathbf{S} \in \mathbf{Aff}(2)$).

Compounding translation and rotation operations, we can get a transformation in the special Euclidean group,

$$\begin{aligned} \mathbf{T} \cdot \mathbf{R} &= \begin{bmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & v_x \\ \sin(\theta) & \cos(\theta) & v_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbf{SE}(2). \end{aligned} \tag{42}$$

Yet, if we add the scaling operation, the compound will belong to the Affine group. One of such compound operator can be written as

$$\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} = \begin{bmatrix} s_x \cos(\theta) & -s_y \sin(\theta) & v_x \\ s_x \sin(\theta) & s_y \cos(\theta) & v_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbf{Aff}(2). \tag{43}$$

When $s_x \neq 0$ and $s_y \neq 0$, the compound operator is invertible. It can be written in form of

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{44}$$

In actual implementation, a high-dimensional relation operator can be represented as a block diagonal matrix in the form of

$$\mathbf{M_r} = \mathbf{diag}(\mathbf{O_{r,1}}, \mathbf{O_{r,2}}, \dots, \mathbf{O_{r,n}}), \tag{45}$$

where $\mathbf{O_{r,i}}$ is the compound operator at the i-th stage. We can multiply $\mathbf{M_r} \cdot \mathbf{v}$ in the following manner,

$$\left[\begin{array}{c|c|c|c} \mathbf{O}_{r,1} & 0 & \dots & 0 \\ \hline 0 & \mathbf{O}_{r,2} & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & \mathbf{O}_{r,n} \end{array}\right] \left[\begin{array}{c} x_1 \\ y_1 \\ \hline x_2 \\ y_2 \\ \hline \vdots \\ \hline x_n \\ y_n \end{array}\right] \tag{46}$$

where $\mathbf{v} = [x_1, y_1, x_2, y_2, \dots, x_n, y_n]^T$ are $2n$ dimensional entity vectors that are split into multiple 2d subspaces.

3.2 CompoundE3D

3.2.1 Translation

Component $\mathbf{T} \in \mathbf{SE}(3)$, illustrated by Figure 4a, is defined as

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{47}$$

3.2.2 Scaling

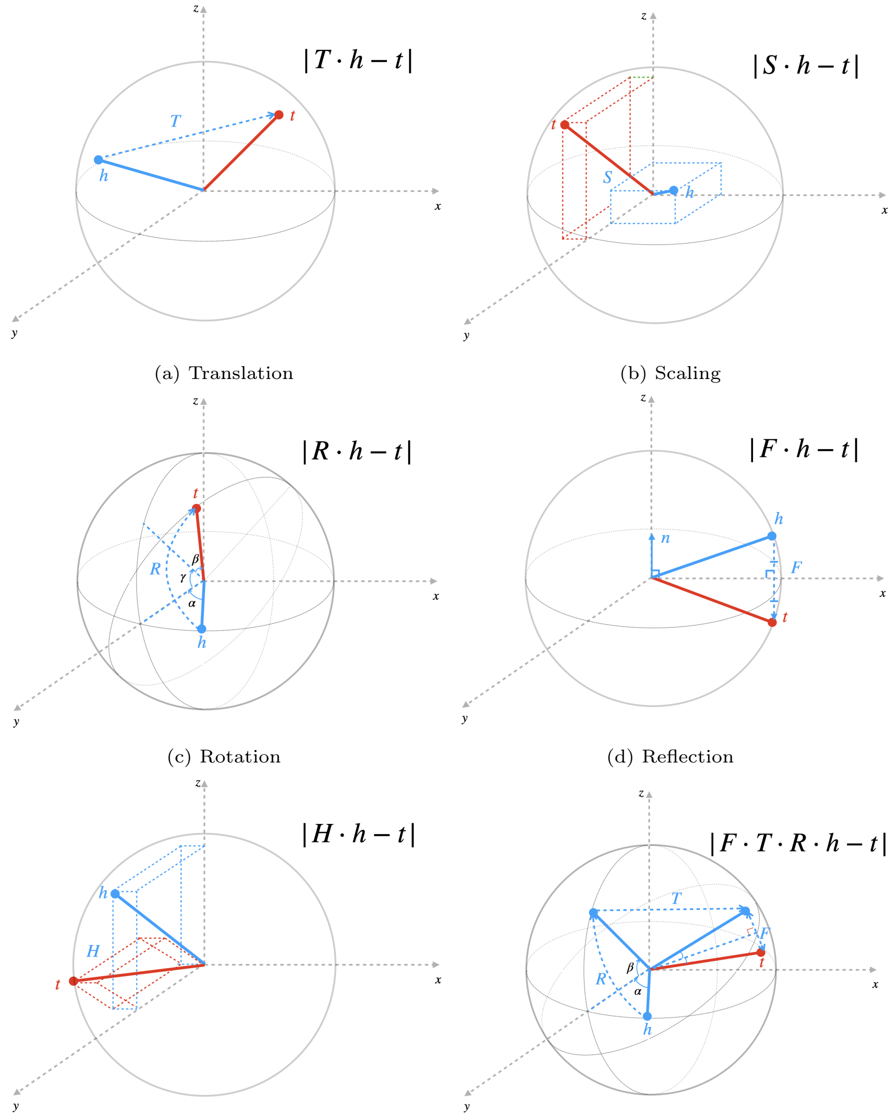
Component $\mathbf{S} \in \mathbf{Aff}(3)$, illustrated by Figure 4b, is defined as

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{48}$$

3.2.3 Rotation

Component $\mathbf{R} \in \mathbf{SO}(3)$, illustrated by Figure 4c, is defined as

$$\mathbf{R} = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma) = \begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{49}$$



(a) Translation (b) Scaling

(c) Rotation (d) Reflection

(e) Shear (f) Compound

Figure 4: Composing different geometric operations in the 3D subspace.

where

$$
\begin{aligned}
a &= \cos(\alpha)\cos(\beta), \\
b &= \cos(\alpha)\sin(\beta)\sin(\gamma) - \sin(\alpha)\cos(\gamma), \\
c &= \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma), \\
d &= \sin(\alpha)\cos(\beta), \\
e &= \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma), \\
f &= \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma), \\
g &= -\sin(\beta), \\
h &= \cos(\beta)\sin(\gamma), \\
i &= \cos(\beta)\cos(\gamma).
\end{aligned} \tag{50}
$$

This general 3D rotation operator is the result of compounding yaw, pitch, and roll rotations. They are, respectively, defined as

- Yaw rotation component:

$$
\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{51}
$$

- Pitch rotation component:

$$
\mathbf{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{52}
$$

- Roll rotation component:

$$
\mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) & 0 \\ 0 & \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{53}
$$

3.2.4 Reflection

Component $\mathbf{F} \in \mathbf{SO}(3)$, illustrated by Figure 4d, is defined as

$$
\mathbf{F} = \begin{bmatrix} 1 - 2n_x^2 & -2n_x n_y & -2n_x n_z & 0 \\ -2n_x n_y & 1 - 2n_y^2 & -2n_y n_z & 0 \\ -2n_x n_z & -2n_y n_z & 1 - 2n_z^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{54}
$$

The above expression is derived from the Householder reflection, $\mathbf{F} = \mathbf{I} - \mathbf{2nn^T}$. In the 3D space, $\mathbf{n}$ is a 3-D unit vector that is perpendicular to the reflecting hyper-plane, $\mathbf{n} = [n_x, n_y, n_z]$.

3.2.5 Shear

Component $\mathbf{H} \in \mathbf{Aff}(3)$, illustrated by Figure 4e, is defined as

$$\mathbf{H} = \mathbf{H}_{yz}\mathbf{H}_{xz}\mathbf{H}_{xy} = \begin{bmatrix} 1 & \mathrm{Sh}_x^y & \mathrm{Sh}_x^z & 0 \\ \mathrm{Sh}_y^x & 1 & \mathrm{Sh}_y^z & 0 \\ \mathrm{Sh}_z^x & \mathrm{Sh}_z^y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{55}$$

The shear operator is the result of compounding 3 operators: $\mathbf{H}_{yz}$, $\mathbf{H}_{xz}$, and $\mathbf{H}_{xy}$ They are mathematically defined as

$$\mathbf{H}_{yz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \mathrm{Sh}_y^x & 1 & 0 & 0 \\ \mathrm{Sh}_z^x & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{56}$$

$$\mathbf{H}_{xz} = \begin{bmatrix} 1 & \mathrm{Sh}_x^y & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \mathrm{Sh}_z^y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{57}$$

$$\mathbf{H}_{xy} = \begin{bmatrix} 1 & 0 & \mathrm{Sh}_x^z & 0 \\ 0 & 1 & \mathrm{Sh}_y^z & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{58}$$

Matrix $\mathbf{H}_{yz}$ has a physical meaning - the shear transformation that shifts the y- and z- components by a factor of the x component. Similar physical interpretations are applied to $\mathbf{H}_{xz}$ and $\mathbf{H}_{xy}$.

The above transformations can be cascaded to yield a compound operator; e.g.,

$$\mathbf{O} = \mathbf{H} \cdot \mathbf{S} \cdot \mathbf{R} \cdot \mathbf{F} \cdot \mathbf{T}, \tag{59}$$

In the actual implementation, we use the operator's representation in regular Cartesian coordinates instead of the homogeneous coordinate. Furthermore, a high-dimensional relation operator can be represented as a block diagonal matrix in the form of

$$\mathbf{M_r} = \mathbf{diag}(\mathbf{O_{r,1}}, \mathbf{O_{r,2}}, \ldots, \mathbf{O_{r,n}}), \tag{60}$$

where $\mathbf{O_{r,i}}$ is the compound operator at the i-th stage. We can multiply $\mathbf{M_r} \cdot \mathbf{v}$ in the following manner,

$$\left[\begin{array}{c|c|c|c} \mathbf{O}_{r,1} & 0 & \ldots & 0 \\ \hline 0 & \mathbf{O}_{r,2} & \ldots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \ldots & \mathbf{O}_{r,n} \end{array}\right] \left[\begin{array}{c} x_1 \\ y_1 \\ z_1 \\ \hline x_2 \\ y_2 \\ z_2 \\ \hline \vdots \\ \hline x_n \\ y_n \\ z_n \end{array}\right] \tag{61}$$

where $\mathbf{v} = [x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n]^T$ are $3n$ dimensional entity vectors that are split into multiple 3d subspaces.

3.3 Connections to other KGEs

Since CompoundE is a general model that includes different transformation components, it can be reduced to several famous KGE models.

Derivation of TransE. We begin with CompoundE-Head and set its rotation component to identity matrix $\mathbf{I}$ and scaling parameters to $\mathbf{1}$. Then, we get the scoring function of TransE as,

$$f_r(h,t) = \|\mathbf{T_r} \cdot \mathbf{I} \cdot \mathbf{I} \cdot \mathbf{h} - \mathbf{t}\| = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|. \tag{62}$$

Derivation of RotatE [94]. We can derive the scoring function of RotatE from CompoundE-Head by setting the translation component to $\mathbf{I}$ (translation vector $\mathbf{t} = \mathbf{0}$) and scaling component to $\mathbf{1}$,

$$f_r(h,t) = \|\mathbf{I} \cdot \mathbf{R_r} \cdot \mathbf{I} \cdot \mathbf{h} - \mathbf{t}\| = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|. \tag{63}$$

Derivation of LinearRE [78]. We can add back the translation component for the head transformation,

$$f_r(h,t) = \|\mathbf{T_r} \cdot \mathbf{I} \cdot \mathbf{S_r} \cdot \mathbf{h} - \mathbf{I} \cdot \mathbf{I} \cdot \hat{\mathbf{S}}_\mathbf{r} \cdot \mathbf{t}\| = \|\mathbf{h} \odot \mathbf{r^H} + \mathbf{r} - \mathbf{t} \odot \mathbf{r^T}\|. \tag{64}$$

Derivation of PairRE [14]. CompoundE-Complete can be reduced to PairRE by setting both translation and rotation component to $\mathbf{I}$, for both head and tail transformation,

$$f_r(h,t) = \|\mathbf{I} \cdot \mathbf{I} \cdot \mathbf{S_r} \cdot \mathbf{h} - \mathbf{I} \cdot \mathbf{I} \cdot \hat{\mathbf{S}}_\mathbf{r} \cdot \mathbf{t}\| = \|\mathbf{h} \odot \mathbf{r^H} - \mathbf{t} \odot \mathbf{r^T}\|. \tag{65}$$

Similarly, we can derive KGE models from CompoundE3D.

Derivation of NagE [132]. Both NagE and CompoundE3D uses $SO(3)$ rotations. CompoundE3D-left can be reduced to NagE by keeping the rotation component, while setting other components to $\mathbf{I}$. Let $\mathbf{R_r} \in SO(3)$,

$$f_r(h,t) = \|\mathbf{I} \cdot \mathbf{I} \cdot \mathbf{R_r} \cdot \mathbf{I} \cdot \mathbf{I} \cdot \mathbf{h} - \mathbf{t}\| = \|\mathbf{R_r} \cdot \mathbf{h} - \mathbf{t}\|. \tag{66}$$

Derivation of ReflectE [140]. CompoundE3D-Complete can be reduced to the full form of ReflectE by keeping the reflection and translation component of the head, and the scaling component of the tail,

$$\begin{aligned} f_r(h,t) &= \|\mathbf{I} \cdot \mathbf{I} \cdot \mathbf{I} \cdot \mathbf{F_r} \cdot \mathbf{T_r} \cdot \mathbf{h} - \mathbf{I} \cdot \hat{\mathbf{S}}_\mathbf{r} \cdot \mathbf{I} \cdot \mathbf{I} \cdot \mathbf{I} \cdot \mathbf{t}\| \\ &= \|\mathbf{M_r^h}(\mathbf{h} + \mathbf{r}') - \mathbf{t_c} \odot \mathbf{r_t}\|. \end{aligned} \tag{67}$$

4 Dataset and Evaluation

4.1 Open Knowledge Graphs and Benchmarking Datasets

Table 4 collects a set of KG that are available for public access. We list these KGs according to the time of their first release. We provide information for the number of entities, the number of facts to indicate the scale and size of each KG. Among these public KGs, WordNet [69] is the oldest and it is first created as a lexical database of English words. Words are connected with semantic relations including Synonymy, Antonymy, Hyponymy, Meronymy, Troponomy, and Entailment. There are similarities and differences between the KGs. For example, both DBpedia [2] and YAGO [93] are derived from information in Wikipedia infoboxes. However, YAGO also includes information from GeoNames that contains spatial information. In addition, DBpedia, YAGO, and Wikidata [104] contain multilingual information. Both ConceptNet [92] and OpenCyc contain a wide range of commonsense concepts and relations.

Table 5 contains commonly used benchmarking datasets. These datasets have different sizes. Each of them covers a different domain. For example, UMLS contains biomedical concepts from the Unified Medical Language System. Similarly, OGB-biokg is also a biomedical KG but with a larger size. Kinship contains relationship between members of a tribe and Countries contains relationships between countries and regions. As indicated by the dataset names, many of them are subsets of the public KGs that are described above. CoDEx is extracted from Wikidata, but presents 3 different versions and each has different degrees and densities. Among these datasets, FB15K-237 and WN18RR are the most adopted datasets for performance benchmarking since inverse relations are removed to avoid the test leakage problem.

Table 4: Major Open Knowledge Graphs.

Name	Year	# Entities	# Facts	Source	Highlights	Access Link
WordNet	1980s	155K	207K	Curated by experts	Lexical database of semantic relations between words	https://wordnet.princeton.edu
ConceptNet	1999	44M	2.4B	Crowdsourced human data and structured feeds	Commonsense concepts and relations	https://conceptnet.io
Freebase	2007	44M	2.4B	Crowdsourced human data and structured feeds	One of the first public KBs/KGs	https://developers.google.com/freebase/
DBpedia	2007	4.3M	70M	Automatically extracted structured data from Wikipedia	Multilingual, Cross-Domain	https://dbpedia.org/sparql
YAGO	2008	10M	120M	Derived from Wikipedia, WordNet, and GeoNames	Multilingual, Fine grained entity types	https://yago-knowledge.org/sparql
Wikidata	2012	50M	500M	Crowdsourced human curation	Collaborative, Multilingual, Structured	https://query.wikidata.org
OpenCyc	2001	2.4M	240k	Created by domain experts	Commonsense concepts and relations	http://www.cyc.com
NELL	2010	-	50M	Extracted from Clueweb09 corpus (1B web pages)	Each fact is given a confidence score. 2.8M beliefs has high confidence score.	http://rtw.ml.cmu.edu/rtw/

Table 5: Knowledge Graph Benchmarking Datasets for Link Prediction.

Dataset	Statistics						Remarks
	#Ent	#Rel	#Train	#Valid	#Test	Avg. Deg.	
Kinship	104	26	8,544	1,068	1,074	82.15	Information about complex relational structure among members of a tribe. Models: NCRL [22], ComplEx-N3-RP [21], RESCAL [76]
UMLS	135	49	5,216	652	661	38.63	Biomedical relationships between categorized concepts of the Unified Medical Language System. Models: TransE [7], DistMult [127], ConvE [24]
Countries	272	2	1,111	24	24	4.35	Relationships between countries, regions, and subregions. Models: RotatE [94], HolE [75]
FB15K	14,951	1,345	483,142	50,000	59,071	13.2	A subset of the Freebase. Textual descriptions of entities are available. Models: AutoSF [144], ComplEx-N3 [3], QuatE [142]
FB15K-237	14,951	237	272,115	17,535	20,466	19.74	Derived from FB-15K by removing inverse relations to avoid test leakage problem. Models: NBFNet [152], ComplEx-DURA [146], MEIM [99]
WN18	40,943	18	141,442	5,000	5,000	1.2	A subset of the WordNet. Models: LinearRE [78], DensE [66], TuckER [4]

Table 5: Continued.

Dataset	Statistics						Remarks
	#Ent	#Rel	#Train	#Valid	#Test	Avg. Deg.	
WN18RR	40,943	11	86,835	3,034	3,134	2.19	Derived from WN18 by removing inverse relations to avoid test leakage problem. Models: SimKGC [108], LMKE [112], StAR [106]
YAGO3-10	123,182	37	1,079,040	5,000	5,000	9.6	Subset of YAGO3 (extension of YAGO) that contains entities associated with at least 10 different relations, describing citizenship, gender, and profession of people. Models: BoxE [1], HAKE[147], InteractE [102]
DB100K	99,604	470	597,572	50,000	50,000	12	A subset of the DBpedia. Each entity appears in at least 20 different relations Models: AcrE [82], TransHER [60], SEEK [126]
CoDEx-S	2,034	42	32,888	1,827	1,828	21.47	Extracted from Wikidata. Each entity has degree at least 15. Hard negative samples are provided. Models: ACTC [87]
CoDEx-M	17,050	51	185,584	10,310	10,311	13.45	Extracted from Wikidata. Each entity has degree at least 10. Hard negative samples are provided. Models: ComplEx-N3-RP [21]
CoDEx-L	77,951	69	551,193	30,622	30,622	25.62	Extracted from Wikidata. Each entity has degree at least 5. Models: PIE [15]

Table 5: Continued.

Dataset	Statistics						Remarks
	#Ent	#Rel	#Train	#Valid	#Test	Avg. Deg.	
OGB-Wikikg2	2,500,604	535	16,109k	429k	598k	8.79	Extracted from Wikidata. Triple split based on timestamp rather than random split. Models: PairRE [14], NodePiece [32], Rot-Pro [91]
OGB-Biokg	93,773	51	4,763k	163k	163k	47.5	Created from a large number of biomedical data repositories. Contains information about diseases, proteins, drugs, side effects, and protein functions. Models: KGBench [145]

4.2 Evaluation Metrics and Leaderboard

The link prediction performance of KGE models is typically evaluated using the Mean Reciprocal Rank (MRR) and Hits@k metrics. The MRR is the average of the reciprocal ranks of the ground truth entities. The Hits@k is the fraction of test triples for which the ground truth entity is ranked among the top k candidates. The MRR and Hits@k metrics are defined as follows:

- The MRR is calculated as:

$$\text{MRR} = \frac{1}{|D|} \sum_{i \in D} \frac{1}{\text{Rank}_i}, \tag{68}$$

  where $|D|$ is the number of test triples, and Rank_i is the rank of the ground truth entity in the list of top candidates for the ith test triple.

- The Hits@k is calculated as:

$$\text{Hits@}k = \frac{1}{|D|} \sum_{i \in D} \mathbb{1}\{\text{Rank}_i \leq k\}, \tag{69}$$

  where $\mathbb{1}\{\cdot\}$ is the indicator function.

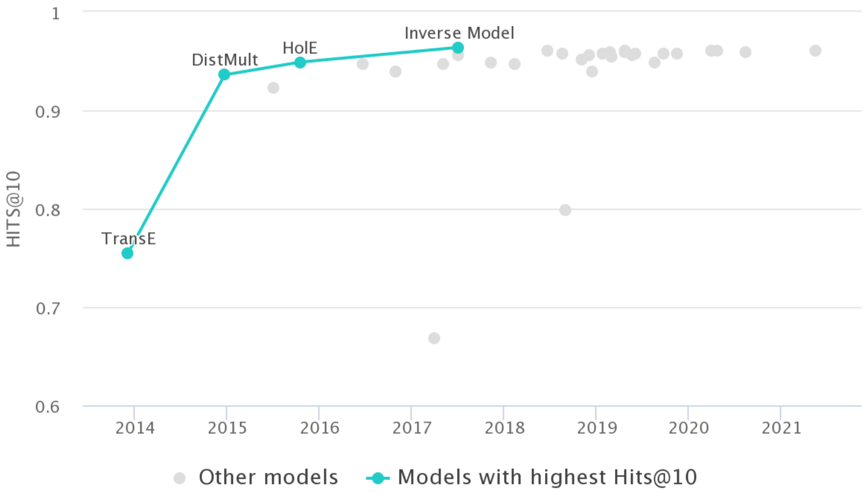
Higher MRR and Hits@k values indicate better model performance. This is because they mean that the model is more likely to rank the ground truth entity higher in the list of top candidates, and to rank it among the top k candidates, respectively. In order to prevent the model from simply memorizing the triples in the KG and ranking them higher, the filtered rank is typically used to evaluate the link prediction performance of KGE models. The filtered rank is the rank of the ground truth entity in the list of top candidates, but only considering candidates that would result in unseen triples. In addition to the MRR and Hits@k metrics, other evaluation metrics for KG completion include Mean Average Precision (MAP), Precision@k, etc. The choice of evaluation metric depends on the specific application. For example, if the goal is to rank the top entities for a given triple, then the MRR or Hits@k metrics may be more appropriate. If the goal is to find all entities that are related to a given entity, then the MAP or Precision@k metrics may be more appropriate.

We also show a comparison of the Hits@10 performance of recently published works for link prediction in Figure 5. The figure includes the results on the FB15k, FB15k-237, WN18, WN18RR, and YAGO3-10 datasets. While KGE models such as MEIM [99] remain competitive, many of the pretrained language model (PLM) approaches such as SimKGC [108] and LMKE [112] top the leaderboards. In the next section, we will discuss this emerging trend of using PLM for KG completion.

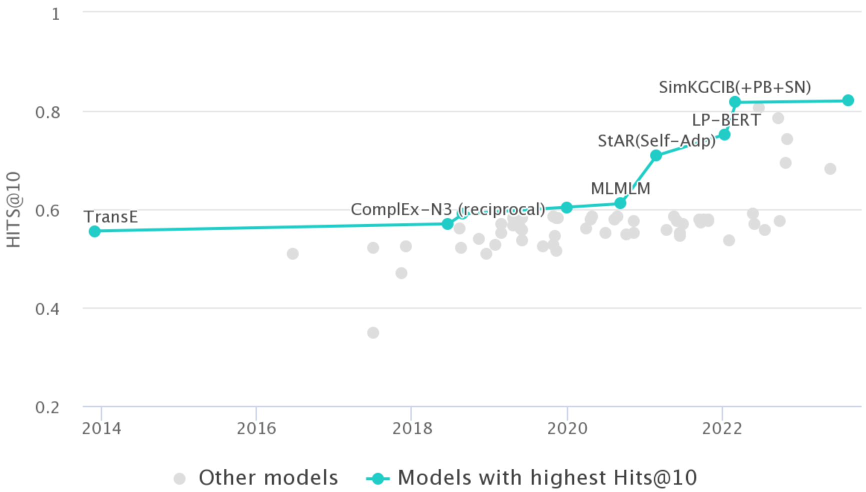
(a) FB15K

(b) FB15K-237

Figure 5: HITS@10 score of previous KGE models for datasets.
Source: https://paperswithcode.com/sota

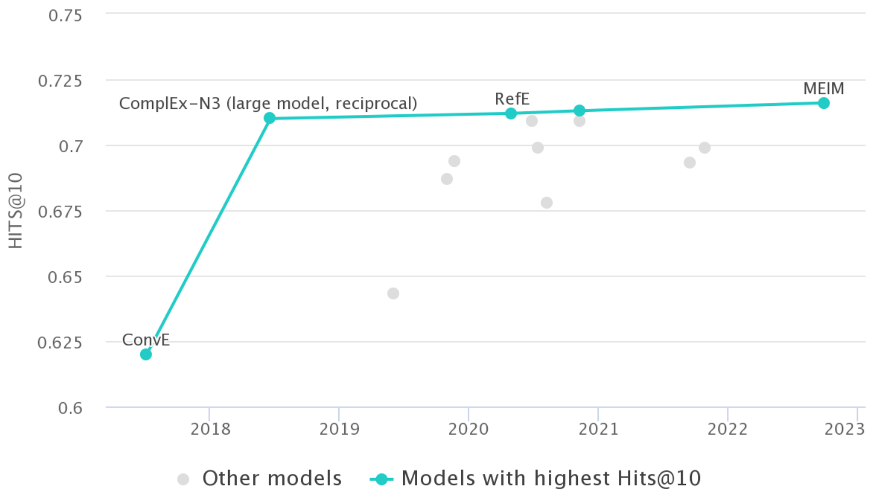


(c) WN18



(d) WN18RR

Figure 5: HITS@10 score of previous KGE models for datasets (cont.).
Source: https://paperswithcode.com/sota



(e) YAGO3-10

Figure 5: HITS@10 score of previous KGE models for datasets (cont.)
Source: https://paperswithcode.com/sota

5 Emerging Direction

5.1 Neural Network Models for Knowledge Graph Completion

Before discussing the PLM approach, it is worthwhile to introduce neural network models for KG completion since PLMs also belong to this line of approach and the logic for training and inference between these models are similar. a Multilayer Perceptron (MLP) [26] is used to measure the likelihood of unseen triples for link prediction. NTN [90] adopts a bilinear tensor neural layer to model interactions between entities and relations of triples. ConvE [24] reshapes and stacks the head entity and the relation vector to form a 2D shape data, applies Convolutional Neural Networks (CNNs) to extract features, and uses extracted features to interact with tail embedding. R-GCN [86] applies a Graph Convolutional Network (GCN) and considers the neighborhood of each entity equally. CompGCN [103] performs a composition operation over each edge in the neighborhood of a central node. The composed embeddings are then convolved with specific filters representing the original and the inverse relations, respectively. KBGAN [10] optimizes a generative adversarial network to generate the negative samples. KBGAT [70] applies graph attention networks to capture both entity and relation features in any given entity's neighborhood. ConvKB [23] applies 1D convolution on stacked entity and relation embeddings of a triple to extract feature maps and applies a nonlinear classifier to predict the likelihood of the triple. Structure-Aware Convolutional Network (SACN) [88] uses a weighted-GCN encoder and a Conv-TransE decoder to extract the embedding. This synergy successfully leverages graph connectivity structure information. InteractE [102] uses network design ideas including feature permutation, a novel feature reshaping, and circular convolution compared to ConvE and outperforms baseline models significantly. ParamE [16] uses neural networks instead of relation embedding to model the relational interaction between head and tail entities. MLP, CNN, and gated structure layers are experimented and the gated layer turns out to be far more effective than embedding approaches. ReInceptionE [120] applies Inception network to increase the interactions between head and relation embeddings. Relation-aware attention mechanism in the model aggregates the local neighborhood features and the global entity information. M-DCN [148] adopts a multi-scale dynamic convolutional network to model complex relations such as 1-N, N-1, and N-N relations. A related model called KGBoost is a tree classifier-based method that proposes a novel negative sampling method and uses the XGBoost classifier for link prediction. GreenKGC [114] is a modularized KGC method inspired by Discriminant Feature Learning (DFT) [56, 133], which extracts the most discriminative feature from trained embeddings for binary classifier learning.

5.2 Pretrained Language Models for Knowledge Graph Completion

With the advent of large language models (LLM) in recent years, more and more NLP tasks are improved significantly improved by pretrained transformer-based models. In recent years, researchers have also started to think about using transformer-based models as solutions for KG-related tasks. A general illustration of the transformer-based approach for KG completion is shown in Figure 6. However, initial results from early papers have not fully demonstrated the effectiveness of language model-based solutions. It requires not only significantly more computational resources for training than KGE models, but also has slow inference speed. There are still many issues with PLM approach that are yet to be solved.

The key difference between traditional KGE approach and PLM-based approach is that the former focuses on local structural information in graphs, whereas the latter relies on PLMs to decide the contextual relatedness between entities' names and descriptions. Textual descriptions of entities are usually available in many KGs such as Freebase, WordNet, and Wikidata. Triples are created from a large amount of corpus on entity description through information extraction. These entity descriptions are often stored in the knowledge base together with the entity entry. These textual descriptions are very useful information and one can use PLMs to extract textual features from

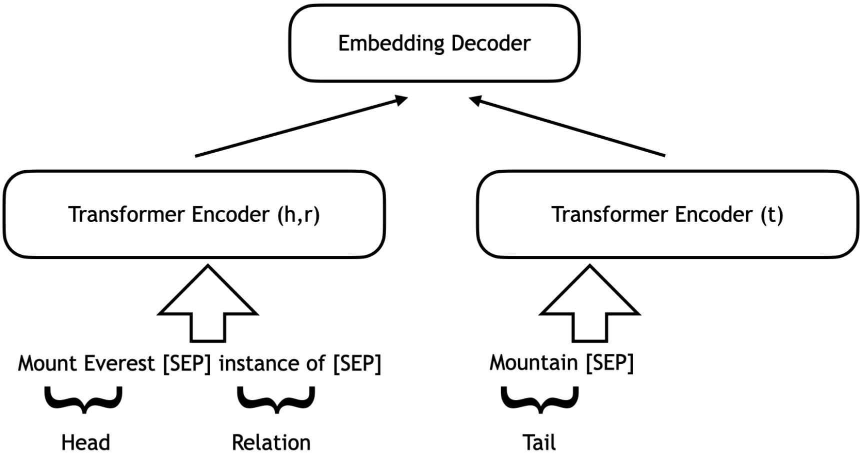


Figure 6: An Illustrative example of the transformer-based approach for KG completion.

these descriptions. PLMs are transformer-based models with many model parameters that are trained over large-scale corpora. PLMs are known to be able to generate good features for various NLP tasks. One of the famous PLMs is BERT [25]. BERT is a pretrained bidirectional language model that is built based on transformer architecture. Since one of the tasks of BERT pretraining is next-sentence prediction, it naturally generates good features for characterizing whether 2 sentences are closely related. KG-BERT [134] is one of the first models we know of that uses PLMs to extract linguistic features from entity descriptions. It leverages the advantage of BERT, which is trained using next-sentence prediction to determine the association of head entity and tail entity descriptions for link prediction, and also triple classification tasks using the same logic.

ERNIE [149] proposes a transformer-based architecture that leverages lexical, syntactic, and knowledge information simultaneously by encoding textual description tokens through cascaded transformer blocks, as well as concatenating textual features and entity embedding features to achieve information fusion. K-BERT alleviates the knowledge noise issue by introducing soft positions and a visible matrix. StAR [106] and SimKGC [108] both use two separate transformer encoders to extract the textual representation of (head entity, relation) and (tail entity). However, they adopt very different methods to model the interaction between two encodings. StAR learns from previous NLP literature [8, 81] to apply interactive concatenation of features such as multiplication, subtraction, and the embedding vectors themselves to represent the semantic relationship between two parts of triples. On the other hand, SimKGC computes cosine similarity between two textual encodings.

SimKGC also proposes new negative sampling methods including in-batch negatives, pre-batch negatives, and self negatives to improve the performance. Graph structural information is also considered in SimKGC to boost the score of entities that appear in the K-hop neighborhood. KEPLER [111] uses the textual descriptions of head and tail entities as initialization for entity embeddings and uses TransE embedding as a decoder. The masked language modeling (MLM) loss is added to the knowledge embedding (KE) loss for overall optimization. InductivE [105] uses features from pretrained BERT as graph embedding initialization for inductive learning on commonsense KGs (CKG). Experimental results on CKG show that fastText features can exhibit comparable performance as that from BERT. BERTRL [138] fine-tunes PLM by using relation instances and possible reasoning paths in the local neighborhood as training samples. Relation paths in the local neighborhood are known to carry useful information for predicting direct relations between two entities [57, 89, 122]. In BERTRL, each relation path is linearized to a sequence of tokens. The target triple and linearized relation paths are each fed to a pretrained BERT model to produce a likelihood score. The final link prediction decisions are made by aggregating these scores. The basic logic in this work is to perform link prediction through the relation path around the target triple. KGT5 [85] propose to use a popular seq2seq transformer model named T5 [80] to pretrain on the link prediction task and perform KGQA. During link prediction training, a textual signal “predict tail” or “predict head” is prepended to the concatenated entity and relation sequence that’s divided by a separation token. This sequence is fed to the encoder and the decoder’s objective is to autoregressively predict the corresponding tail or head based on the textual signal. To perform question answering, textual signal “predict answer” is prepended to the query and we expect the decoder to autoregressively generate a corresponding answer to the query. This approach claims to significantly reduce the model size and inference time compared to other models. PKGC [68] proposes a new evaluation metric that is more accurate under an open-world assumption (OWA) setting. More recently, TAGREAL [52] uses text mining techniques to mine optimal prompts from PLM-associated huge corpus, and probe more accurate knowledge from PLM.

6 Conclusion

In conclusion, this paper has provided a comprehensive overview of the current state of research in KGE. We have explored the evolution of KGE models, with a particular focus on two main branches: distance-based methods and semantic matching-based methods. Through our analysis, we have uncovered intriguing connections among recently proposed models and identified a promising trend that combines geometric transformations to enhance the performance of existing KGE models. Moreover, this paper has curated valuable resources for KG research, including survey papers, open KGs, benchmarking datasets, and leaderboard results for link prediction. We have also delved into emerging directions that leverage neural network models, including graph neural networks and PLM, and highlighted how these approaches can be integrated with embedding-based models to achieve improved performance on diverse downstream tasks. In the rapidly evolving field of KG completion, this paper serves as a valuable reference for researchers and practitioners, offering insights into the past developments, current trends, and potential future directions. By providing a unified framework in the form of CompoundE and CompoundE3D, we aim to inspire further innovation in KGE methods and facilitate the construction of more accurate and comprehensive KGs. As the demand for knowledge-driven applications continues to grow, the pursuit of effective KGE models remains a pivotal area of research, and this paper lays the groundwork for future advancements in the field.

7 Acknowledgments

The authors acknowledge the Center for Advanced Research Computing (CARC) at the University of Southern California for providing computing resources that have contributed to the research results reported within this publication. URL: https://carc.usc.edu.

References

[1] R. Abboud, I. Ceylan, T. Lukasiewicz, and T. Salvatori, "Boxe: A box embedding model for knowledge base completion", *Advances in Neural Information Processing Systems*, 33, 2020, 9649–61.

[2] 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*, 2007, 722–35.

[3] I. Balažević, C. Allen, and T. Hospedales, "Multi-relational Poincaré Graph Embeddings", in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Vol. 32, 2019.

[4] I. Balažević, C. Allen, and T. Hospedales, "TuckER: Tensor Factorization for Knowledge Graph Completion", in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, 5185–94.

[5] A. Bastos, K. Singh, A. Nadgeri, S. Shekarpour, I. O. Mulang, and J. Hoffart, "Hopfe: Knowledge graph representation learning using inverse hopf fibrations", in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, 89–99.

[6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives", *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 2013, 1798–828.

[7] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data", *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2013, 2787–95.

[8] S. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference", in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, 632–42.

[9] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications", *IEEE transactions on knowledge and data engineering*, 30(9), 2018, 1616–37.

[10] L. Cai and W. Y. Wang, "KBGAN: Adversarial Learning for Knowledge Graph Embeddings", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, 1470–80.

[11] J. Cao, J. Fang, Z. Meng, and S. Liang, "Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces", *arXiv preprint arXiv:2211.03536*, 2022.

[12] Z. Cao, Q. Xu, Z. Yang, X. Cao, and Q. Huang, "Dual quaternion knowledge graph embeddings", in *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021, 6894–902.

[13] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré, "Low-Dimensional Hyperbolic Knowledge Graph Embeddings", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, 6901–14.

[14] L. Chao, J. He, T. Wang, and W. Chu, "PairRE: Knowledge Graph Embeddings via Paired Relation Vectors", in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, 4360–9.

[15] L. Chao, X. Lin, T. Wang, and W. Chu, "PIE: a parameter and inference efficient solution for large scale knowledge graph embedding reasoning", *arXiv preprint arXiv:2204.13957*, 2022.

[16] F. Che, D. Zhang, J. Tao, M. Niu, and B. Zhao, "Parame: Regarding neural network parameters as relation embeddings for knowledge graph completion", in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 03, 2020, 2774–81.

[17] K. Chen, Y. Wang, Y. Li, and A. Li, "RotateQVS: Representing Temporal Information as Rotations in Quaternion Vector Space for Temporal Knowledge Graph Completion", in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, 5843–57.

[18] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment", *arXiv preprint arXiv:1611.03954*, 2016.

[19] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph", *Expert Systems with Applications*, 141, 2020, 112948.

[20] Y. Chen, J. Liu, Z. Zhang, S. Wen, and W. Xiong, "MöbiusE: Knowledge Graph Embedding on Möbius Ring", *Knowledge-Based Systems*, 227, 2021, 107181.

[21] Y. Chen, P. Minervini, S. Riedel, and P. Stenetorp, "Relation Prediction as an Auxiliary Training Objective for Improving Multi-Relational Graph Representations", in *Proceedings of the 3rd Conference on Automated Knowledge Base Construction*, 2021.

[22] K. Cheng, N. Ahmed, and Y. Sun, "Neural Compositional Rule Learning for Knowledge Graph Reasoning", in *The Eleventh International Conference on Learning Representations*, 2022.

[23] T. D. N. Dai Quoc Nguyen, D. Q. Nguyen, and D. Phung, "A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network", in *Proceedings of NAACL-HLT*, 2018, 327–33.

[24] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings", in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018, 1811–8.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv:1810.04805*, 2018.

[26] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion", in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, 601–10.

[27] Z. Du, C. Zhou, J. Yao, T. Tu, L. Cheng, H. Yang, J. Zhou, and J. Tang, "CogKR: Cognitive Graph for Multi-hop Knowledge Reasoning", *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[28] T. Ebisu and R. Ichise, "Generalized translation-based embedding of knowledge graph", *IEEE Transactions on Knowledge and Data Engineering*, 32(5), 2019, 941–51.

[29] T. Ebisu and R. Ichise, "Toruse: Knowledge graph embedding on a lie group", in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[30] M. Fan, Q. Zhou, E. Chang, and F. Zheng, "Transition-based knowledge graph embedding with relational mapping properties", in *Proceedings of the 28th Pacific Asia conference on language, information and computing*, 2014, 328–37.

[31] J. Feng, M. Huang, M. Wang, M. Zhou, Y. Hao, and X. Zhu, "Knowledge graph embedding by flexible translation", in *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2016.

[32] M. Galkin, E. Denis, J. Wu, and W. L. Hamilton, "NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs", in *International Conference on Learning Representations*, 2021.

[33] C. Gao, C. Sun, L. Shan, L. Lin, and M. Wang, "Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding", in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, 385–94.

[34] A. Garcia-Durán, A. Bordes, and N. Usunier, "Composing relationships with translations", *PhD thesis*, CNRS, Heudiasyc, 2015.

[35] C. Ge, X. Liu, L. Chen, B. Zheng, and Y. Gao, "LargeEA: Aligning Entities for Large-scale Knowledge Graphs", *Proceedings of the VLDB Endowment*, 15(2), 2022, 237–45.

[36] X. Ge, Y. C. Wang, B. Wang, C.-C. J. Kuo, *et al.*, "TypeEA: Type-Associated Embedding for Knowledge Graph Entity Alignment", *APSIPA Transactions on Signal and Information Processing*, 12(1),

[37] X. Ge, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, "CompoundE: Knowledge graph embedding with translation, rotation and scaling compound operations", *arXiv preprint arXiv:2207.05324*, 2022.

[38] X. Ge, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, "Knowledge Graph Embedding with 3D Compound Geometric Transformations", *arXiv preprint arXiv:2304.00378*, 2023.

[39] X. Ge, Y.-C. Wang, B. Wang, and C. J. Kuo, "CORE: A knowledge graph entity type prediction method via complex space regression and embedding", *Pattern Recognition Letters*, 157, 2022, 97–103.

[40] N. Guan, D. Song, and L. Liao, “Knowledge graph embedding with concepts”, *Knowledge-Based Systems*, 164, 2019, 38–44.
[41] J. Guo and S. Kok, “BiQUE: Biquaternionic Embeddings of Knowledge Graphs”, *arXiv preprint arXiv:2109.14401*, 2021.
[42] C. Gutiérrez and J. F. Sequeda, “Knowledge graphs”, *Communications of the ACM*, 64(3), 2021, 96–104.
[43] S. He, K. Liu, G. Ji, and J. Zhao, “Learning to represent knowledge graphs with gaussian embedding”, in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, 623–32.
[44] E. Hoffer and N. Ailon, “Deep metric learning using triplet network”, in *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, Springer, 2015, 84–92.
[45] A. Hogan, E. Blomqvist, M. Cochez, C. d’Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, *et al.*, “Knowledge graphs”, *ACM Computing Surveys (Csur)*, 54(4), 2021, 1–37.
[46] H. Huang, G. Long, T. Shen, J. Jiang, and C. Zhang, “RatE: Relation-adaptive translating embedding for knowledge graph completion”, *arXiv preprint arXiv:2010.04863*, 2020.
[47] X. Huang, J. Tang, Z. Tan, W. Zeng, J. Wang, and X. Zhao, “Knowledge graph embedding by relational and entity rotation”, *Knowledge-Based Systems*, 229, 2021, 107310.
[48] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, “Knowledge graph embedding via dynamic mapping matrix”, in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, 687–96.
[49] G. Ji, K. Liu, S. He, and J. Zhao, “Knowledge graph completion with adaptive sparse transfer matrix”, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, 985–91.
[50] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, “A survey on knowledge graphs: Representation, acquisition, and applications”, *IEEE transactions on neural networks and learning systems*, 33(2), 2021, 494–514.
[51] Y. Jia, Y. Wang, H. Lin, X. Jin, and X. Cheng, “Locally adaptive translation for knowledge graph embedding”, in *Thirtieth AAAI conference on artificial intelligence*, 2016.
[52] P. Jiang, S. Agarwal, B. Jin, X. Wang, J. Sun, and J. Han, “Text-Augmented Open Knowledge Graph Completion via Pre-Trained Language Models”, *arXiv preprint arXiv:2305.15597*, 2023.

[53] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs", *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 31, 2018, 4289–300.

[54] M. Kejriwal, "Knowledge graphs: A practical review of the research landscape", *Information*, 13(4), 2022, 161.

[55] P. Kolyvakis, A. Kalousis, and D. Kiritsis, "Hyperbolic knowledge graph embeddings for knowledge base completion", in *European Semantic Web Conference*, Springer, 2020, 199–214.

[56] C.-C. J. Kuo and A. M. Madni, "Green learning: Introduction, examples and outlook", *J. Vis. Commun. Image. Represent.*, 2022, 103685.

[57] N. Lao, T. Mitchell, and W. Cohen, "Random walk inference and learning in a large scale knowledge base", in *Proceedings of the 2011 conference on empirical methods in natural language processing*, 2011, 529–39.

[58] J. Li and Y. Yang, "STaR: Knowledge Graph Embedding by Scaling, Translation and Rotation", in *International Conference on AI and Mobile Services*, 2022, 31–45.

[59] R. Li, J. Zhao, C. Li, D. He, Y. Wang, Y. Liu, H. Sun, S. Wang, W. Deng, Y. Shen, *et al.*, "House: Knowledge graph embedding with householder parameterization", in *International Conference on Machine Learning*, PMLR, 2022, 13209–24.

[60] Y. Li, W. Fan, C. Liu, C. Lin, and J. Qian, "TranSHER: Translating Knowledge Graph Embedding with Hyper-Ellipsoidal Restriction", in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, 8517–28.

[61] Z. Liang, J. Yang, H. Liu, and K. Huang, "A semantic filter based on relations for knowledge graph completion", in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, 7920–9.

[62] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases", *arXiv preprint arXiv:1506.00379*, 2015.

[63] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion", in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, 2181–7.

[64] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings", in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, 2168–78.

[65] Y. Liu, P. Wang, Y. Li, Y. Shao, and Z. Xu, "AprilE: attention with pseudo residual connection for knowledge graph embedding", in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, 508–18.

[66] H. Lu, H. Hu, and X. Lin, "DensE: An enhanced non-commutative representation for knowledge graph embedding with adaptive semantic hierarchy", *Neurocomputing*, 476, 2022, 115–25.

[67] X. Lv, L. Hou, J. Li, and Z. Liu, "Differentiating concepts and instances for knowledge graph embedding", *arXiv preprint arXiv:1811.04588*, 2018.

[68] X. Lv, Y. Lin, Y. Cao, L. Hou, J. Li, Z. Liu, P. Li, and J. Zhou, "Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach", in, Association for Computational Linguistics, 2022.

[69] G. A. Miller, "WordNet: a lexical database for English", *Communications of the ACM*, 38(11), 1995, 39–41.

[70] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs", in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, 4710–23.

[71] M. Nayyeri, C. Xu, F. Hoffmann, M. M. Alam, J. Lehmann, and S. Vahdati, "Knowledge Graph Representation Learning using Ordinary Differential Equations", in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, 9529–48.

[72] M. Nayyeri, C. Xu, S. Vahdati, N. Vassilyeva, E. Sallinger, H. S. Yazdi, and J. Lehmann, "Fantastic knowledge graph embeddings and how to find the right space for them", in *International Semantic Web Conference*, Springer, 2020, 438–55.

[73] D. Q. Nguyen, K. Sirts, L. Qu, and M. Johnson, "Stranse: a novel embedding model of entities and relationships in knowledge bases", *arXiv preprint arXiv:1606.08140*, 2016.

[74] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs", *Proceedings of the IEEE*, 104(1), 2015, 11–33.

[75] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs", in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30, No. 1, 2016.

[76] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data", in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011, 809–16.

[77] Z. Pan and P. Wang, "Hyperbolic Hierarchy-Aware Knowledge Graph Embedding for Link Prediction", in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, 2941–8.

[78] Y. Peng and J. Zhang, "LineaRE: Simple but Powerful Knowledge Graph Embedding for Link Prediction", in *2020 IEEE International Conference on Data Mining (ICDM)*, IEEE Computer Society, 2020, 422–31.

[79] W. Qian, C. Fu, Y. Zhu, D. Cai, and X. He, "Translating Embeddings for Knowledge Graph Completion with Relation Attention Mechanism.", in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, 4286–92.

[80] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer", *The Journal of Machine Learning Research*, 21(1), 2020, 5485–551.

[81] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, 3982–92.

[82] F. Ren, J. Li, H. Zhang, S. Liu, B. Li, R. Ming, and Y. Bai, "Knowledge Graph Embedding with Atrous Convolution and Residual Learning", in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, 1532–43.

[83] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis", *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2), 2021, 1–49.

[84] A. Sadeghi, D. Graux, H. S. Yazdi, and J. Lehmann, "MDE: multiple distance embeddings for link prediction in knowledge graphs", *arXiv preprint arXiv:1905.10702*, 2019.

[85] A. Saxena, A. Kochsiek, and R. Gemulla, "Sequence-to-Sequence Knowledge Graph Completion and Question Answering", in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[86] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks", in *Proceedings of the European Semantic Web Conference*, 2018, 593–607.

[87] A. Sedova and B. Roth, "ACTC: Active Threshold Calibration for Cold-Start Knowledge Graph Completion", *arXiv preprint arXiv:2305.06395*, 2023.

[88] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion", in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, No. 01, 2019, 3060–7.

[89] Y. Shen, N. Ding, H.-T. Zheng, Y. Li, and M. Yang, "Modeling relation paths for knowledge graph completion", *IEEE Transactions on Knowledge and Data Engineering*, 33(11), 2020, 3607–17.

[90] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion", *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 26, 2013.

[91] T. Song, J. Luo, and L. Huang, "Rot-pro: Modeling transitivity by projection in knowledge graph embedding", *Advances in Neural Information Processing Systems*, 34, 2021, 24695–706.

[92] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge", in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, 4444–51.

[93] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A Core of Semantic Knowledge", in *Proceedings of the 16th International Conference on World Wide Web*, 2007, 697–706.

[94] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space", in *7th International Conference on Learning Representations (ICLR 2019)*, 2019, 1–18.

[95] Z. Tan, X. Zhao, and W. Wang, "Representation learning of large-scale knowledge graphs via entity feature combinations", in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, 1777–86.

[96] Y. Tang, J. Huang, G. Wang, X. He, and B. Zhou, "Orthogonal Relation Transforms with Graph Context Modeling for Knowledge Graph Embedding", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, 2713–22.

[97] Y. Tay, A. T. Luu, and S. C. Hui, "Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs", in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[98] I. Tiddi and S. Schlobach, "Knowledge graphs as tools for explainable machine learning: A survey", *Artificial Intelligence*, 302, 2022, 103627.

[99] H. N. Tran and A. Takasu, "MEIM: Multi-partition Embedding Interaction Beyond Block Term Format for Efficient and Expressive Link Prediction", *arXiv preprint arXiv:2209.15597*, 2022.

[100] H. N. Tran and A. Takasu, "Multi-Partition Embedding Interaction with Block Term Format for Knowledge Graph Completion", in *Proceedings of the European Conference on Artificial Intelligence*, 2020, 833–40, https://arxiv.org/abs/2006.16365.

[101] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction", in *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, 2016, 2071–80.

[102] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. Talukdar, "Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions", in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, Vol. 34, No. 03, 2020, 3009–16.

[103] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based Multi-Relational Graph Convolutional Networks", in *Proceedings of the International Conference on Learning Representations (ICLR) 2020*, 2020.

[104] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledge base", *Communications of the ACM*, 57(10), 2014, 78–85.

[105] B. Wang, G. Wang, J. Huang, J. You, J. Leskovec, and C.-C. J. Kuo, "Inductive learning on commonsense knowledge graph completion", in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, 1–8.

[106] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion", in *Proceedings of the Web Conference 2021*, 2021, 1737–48.

[107] K. Wang, Y. Liu, D. Lin, and M. Sheng, "Hyperbolic Geometry is Not Necessary: Lightweight Euclidean-Based Models for Low-Dimensional Knowledge Graph Embeddings", in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, 464–74.

[108] L. Wang, W. Zhao, Z. Wei, and J. Liu, "SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models", in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, 4281–94.

[109] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications", *IEEE Transactions on Knowledge and Data Engineering*, 29(12), 2017, 2724–43.

[110] S. Wang, K. Fu, X. Sun, Z. Zhang, S. Li, and L. Jin, "Hierarchical-aware relation rotational knowledge graph embedding for link prediction", *Neurocomputing*, 458, 2021, 259–70.

[111] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "KEPLER: A unified model for knowledge embedding and pre-trained language representation", *Transactions of the Association for Computational Linguistics*, 9, 2021, 176–94.

[112] X. Wang, Q. He, J. Liang, and Y. Xiao, "Language models as knowledge embeddings", *arXiv preprint arXiv:2206.12617*, 2022.

[113] Y.-C. Wang, X. Ge, B. Wang, and C.-C. J. Kuo, "AsyncET: Asynchronous Learning for Knowledge Graph Entity Typing with Auxiliary Relations", *arXiv preprint arXiv:2308.16055*, 2023.

[114] Y.-C. Wang, X. Ge, B. Wang, and C.-C. J. Kuo, "Greenkgc: A lightweight knowledge graph completion method", *arXiv preprint arXiv:2208.09137*, 2022.

[115] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes", in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, 1112–9.

[116] J. WESTON, "Support Vector Machines for Multi-Class Pattern Recognition", in *Proc. 7th European Symposium on Artificial Neural Networks*, 1999.

[117] H. Xiao, M. Huang, Y. Hao, and X. Zhu, "Transg: A generative mixture model for knowledge graph embedding", *arXiv preprint arXiv:1509.05488*, 2015.

[118] H. Xiao, M. Huang, and X. Zhu, "From one point to a manifold: Knowledge graph embedding for precise link prediction", *arXiv preprint arXiv:1512.04792*, 2015.

[119] Q. Xie, X. Ma, Z. Dai, and E. Hovy, "An interpretable knowledge transfer model for knowledge base completion", *arXiv preprint arXiv:1704.05908*, 2017.

[120] Z. Xie, G. Zhou, J. Liu, and X. Huang, "ReInceptionE: relation-aware inception network with joint local-global structural information for knowledge graph embedding", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, 5929–39.

[121] B. Xiong, S. Zhu, M. Nayyeri, C. Xu, S. Pan, C. Zhou, and S. Staab, "Ultrahyperbolic knowledge graph embeddings", in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, 2130–9.

[122] W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning", in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, 564–73.

[123] C. Xu and R. Li, "Relation Embedding with Dihedral Group in Knowledge Graph", in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, 263–72.

[124] C. Xu, M. Nayyeri, F. Alkhoury, H. S. Yazdi, and J. Lehmann, "TeRo: A Time-aware Knowledge Graph Embedding via Temporal Rotation", in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, 1583–93.

[125] C. Xu, M. Nayyeri, Y.-Y. Chen, and J. Lehmann, "Knowledge graph embeddings in geometric algebras", *arXiv preprint arXiv:2010.00989*, 2020.

[126] W. Xu, S. Zheng, L. He, B. Shao, J. Yin, and T.-Y. Liu, "SEEK: Segmented Embedding of Knowledge Graphs", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, 3888–97.

[127] B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding Entities and Relations for Learning and Inference in Knowledge Bases", in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.

[128] H. Yang and J. Liu, "Knowledge graph representation learning as groupoid: unifying TransE, RotatE, QuatE, ComplEx", in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, 2311–20.

[129] H. Yang, L. Zhang, B. Wang, T. Yao, and J. Liu, "Cycle or Minkowski: Which is More Appropriate for Knowledge Graph Embedding?", in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, 2301–10.

[130] J. Yang, Y. Shi, X. Tong, R. Wang, T. Chen, and X. Ying, "Improving knowledge graph embedding using affine transformations of entities corresponding to each relation", in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, 508–17.

[131] S. Yang, J. Tian, H. Zhang, J. Yan, H. He, and Y. Jin, "TransMS: Knowledge Graph Embedding for Complex Relations by Multidirectional Semantics.", in *IJCAI*, 2019, 1935–42.

[132] T. Yang, L. Sha, and P. Hong, "NagE: Non-Abelian group embedding for knowledge graphs", in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, 1735–42.

[133] Y. Yang, W. Wang, H. Fu, C.-C. J. Kuo, *et al.*, "On supervised feature selection from high dimensional feature spaces", *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[134] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for knowledge graph completion", *arXiv preprint arXiv:1909.03193*, 2019.

[135] H.-G. Yoon, H.-J. Song, S.-B. Park, and S.-Y. Park, "A translation-based knowledge graph embedding preserving logical property of relations", in *proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, 907–16.

[136] J. Yu, Y. Cai, M. Sun, and P. Li, "MQuadE: a unified model for knowledge fact embedding", in *Proceedings of the 30th International Conference on World Wide Web*, 2021, 3442–52.

[137] J. Yuan, N. Gao, and J. Xiang, "Transgate: knowledge graph embedding with shared gate structure", in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33, No. 01, 2019, 3100–7.

[138] H. Zha, Z. Chen, and X. Yan, “Inductive relation prediction by BERT”, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, No. 5, 2022, 5923–31.

[139] F. Zhang, X. Wang, Z. Li, and J. Li, “TransRHS: A Representation Learning Method for Knowledge Graphs with Relation Hierarchical Structure.”, in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2020, 2987–93.

[140] Q. Zhang, R. Wang, J. Yang, and L. Xue, “Knowledge graph embedding by reflection transformation”, *Knowledge-Based Systems*, 238, 2022, 107861.

[141] Q. Zhang, R. Wang, J. Yang, and L. Xue, “Structural context-based knowledge graph embedding for link prediction”, *Neurocomputing*, 470, 2022, 109–20.

[142] S. Zhang, Y. Tay, L. Yao, and Q. Liu, “Quaternion knowledge graph embeddings”, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, 2735–45.

[143] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, “Interaction embeddings for prediction and explanation in knowledge graphs”, in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, 96–104.

[144] Y. Zhang, Q. Yao, W. Dai, and L. Chen, “AutoSF: Searching scoring functions for knowledge graph embedding”, in *Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE)*, IEEE, 2020, 433–44.

[145] Y. Zhang, Q. Yao, and J. T. Kwok, “Bilinear scoring function search for knowledge graph learning”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2), 2022, 1458–73.

[146] Z. Zhang, J. Cai, and J. Wang, “Duality-induced regularizer for tensor factorization based knowledge graph completion”, *Advances in Neural Information Processing Systems*, 33, 2020, 21604–15.

[147] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, “Learning hierarchy-aware knowledge graph embeddings for link prediction”, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 03, 2020, 3065–72.

[148] Z. Zhang, Z. Li, H. Liu, and N. N. Xiong, “Multi-scale dynamic convolutional network for knowledge graph embedding”, *IEEE Transactions on Knowledge and Data Engineering*, 34(5), 2022, 2335–47.

[149] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, “ERNIE: Enhanced Language Representation with Informative Entities”, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, 1441–51.

[150] X. Zhou, L. Niu, Q. Zhu, X. Zhu, P. Liu, J. Tan, and L. Guo, "Knowledge graph embedding by double limit scoring loss", *IEEE Transactions on Knowledge and Data Engineering*, 34(12), 2021, 5825–39.

[151] X. Zhou, Q. Zhu, P. Liu, and L. Guo, "Learning knowledge embeddings by combining limit-based scoring loss", in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, 1009–18.

[152] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang, "Neural bellman-ford networks: A general graph neural network framework for link prediction", *Advances in Neural Information Processing Systems*, 34, 2021, 29476–90.